RACCOLTA DI GIOCHI E DI UTILITIES PER

# TI99/4A

**GRUPPO EDITORIALE JACKSON** s.r.l.

**DIREZIONE, REDAZIONE E AMMINISTRAZIONE**
Via Rossellini, 12 - 20124 Milano
Telefoni: 68.03.68 - 68.00.54
68.80.951-2-3-4-5
Telex 333436 GEJ IT
SEDE LEGALE: Via G. Pozzone, 55 - 20121 Milano

**DIRETTORE RESPONSABILE:**
Giampietro Zanga

**COORDINAMENTO EDITORIALE:**
A. Cattaneo

**GRAFICA E IMPAGINAZIONE:**
Germani Wilma

**HA COLLABORATO:**
P. Chianura

**FOTOCOMPOSIZIONE:**
GDB fotocomposizione
Via Tagliamento, 4 - Milano
Tel. 56.92.110 - 53.92.546

**STAMPA:**
Grafika 78 - Pioltello - MI

**AUTORIZZAZIONE ALLA PUBBLICAZIONE:**
Trib. di Milano n. 417 del 22-9-'84

**PUBBLICITÀ**
Concessionario per l'Italia e l'Estero
J. Advertising s.r.l.
V.le Restelli, 5
20124 MILANO
Tel. (02)
68.82.895-68.80.606-68.87.233
Tlx 316213 REINA I

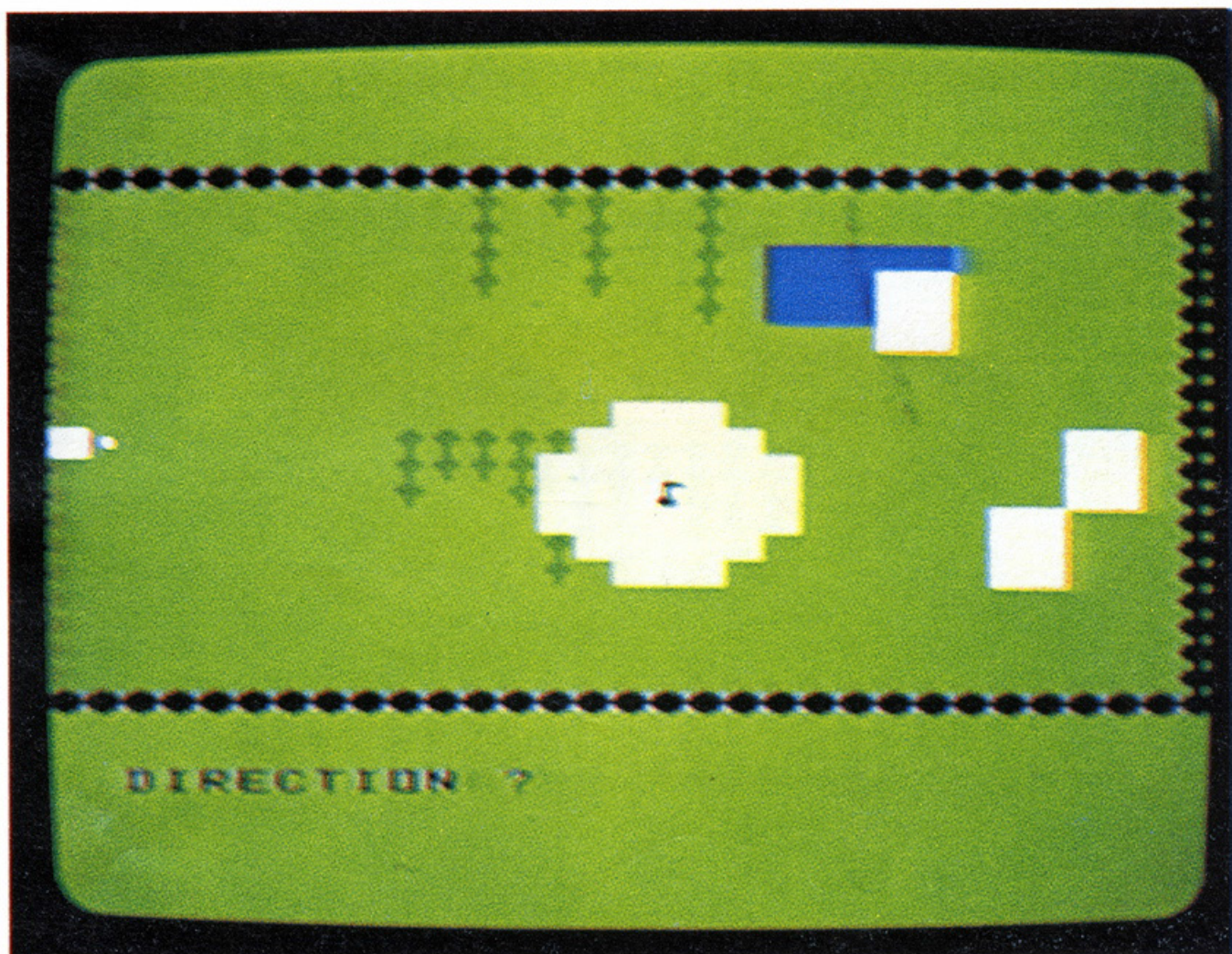
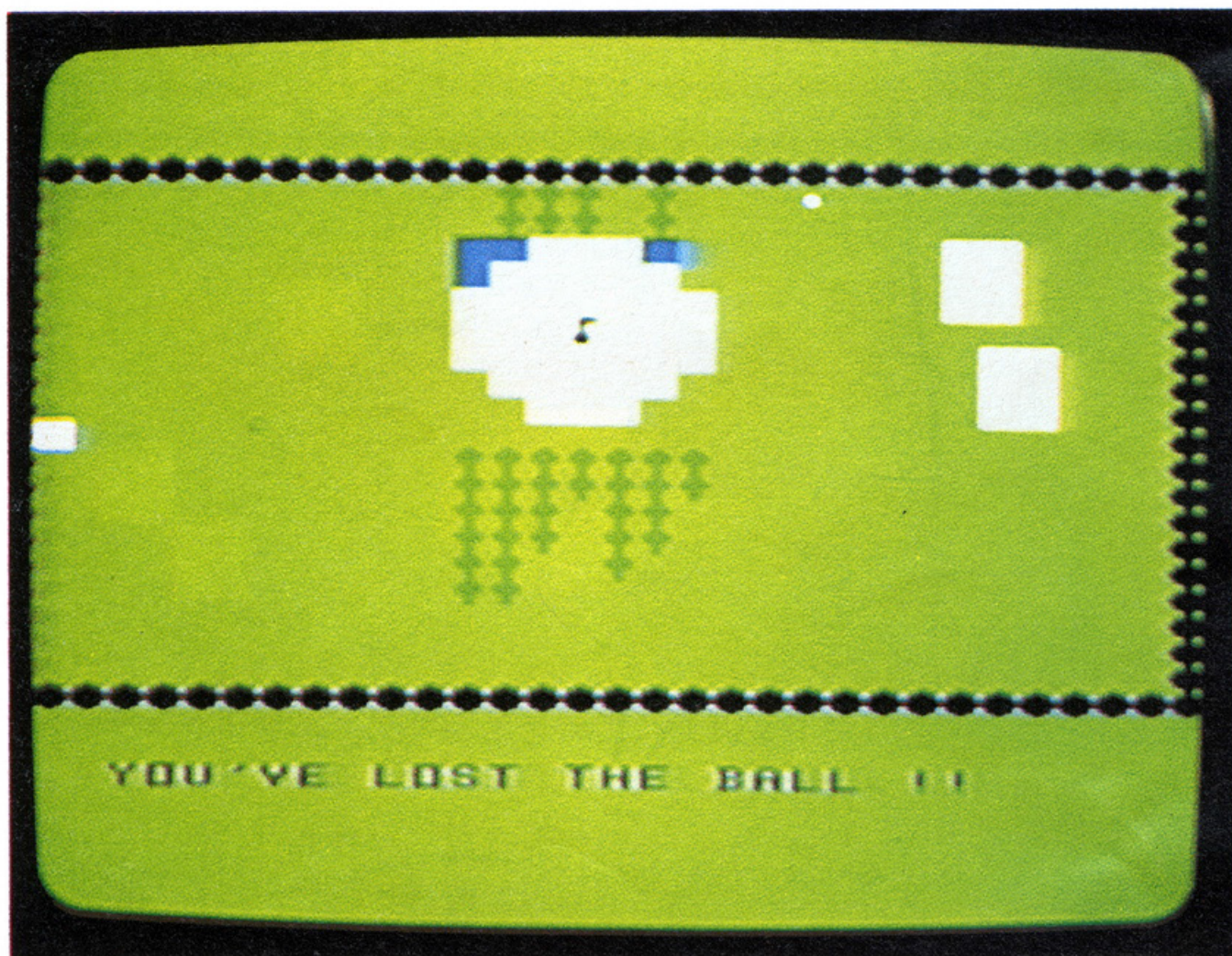
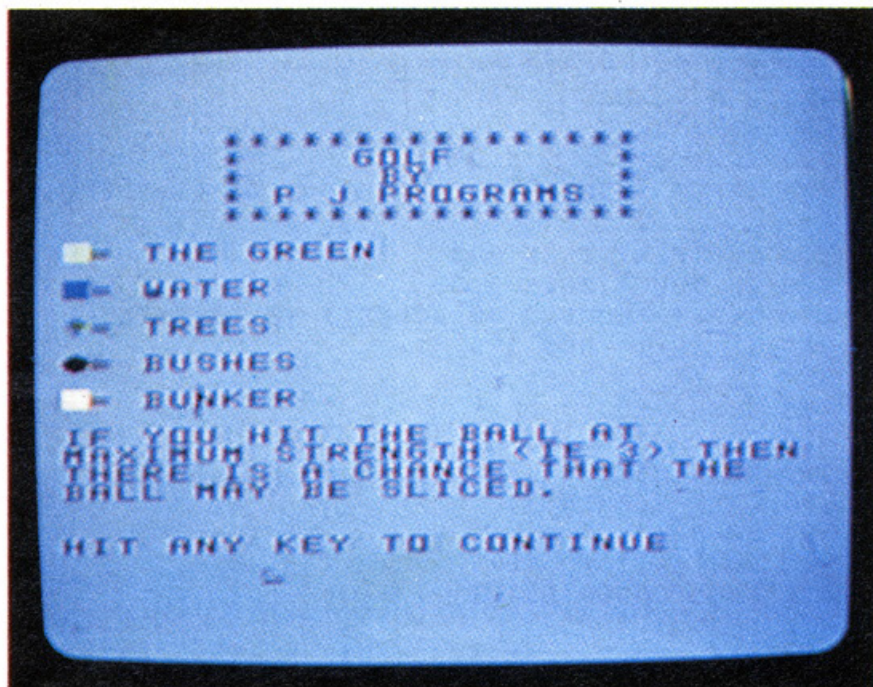
Concessionario esclusivo per la DIFFUSIONE in Italia e Estero:
SODIP - Via Zuretti, 25
20125 MILANO
Spedizione in abbonamento postale Gruppo II/70
Prezzo della rivista L. 6.500
Numeri arretrati L. 13.000



*Con il presente fascicolo prosegue la raccolta dei migliori programmi del TI 99/4A pubblicati su libri e riviste Jackson.*
*In ogni numero un supergame originale inglese descritto nei minimi particolari ed in più una miriade di altri programmi scelti tra utility, grafica, musica, giochi ecc.*
*Ciascun listato è accompagnato da una recensione approfondita e da consigli per un corretto utilizzo.*
*Non solo potrete caricare immediatamente i programmi grazie al master su cassetta allegato alla confezione, ma anche personalizzarli servendovi dei listati e della descrizione pubblicata all'interno del fascicolo.*
*La Jackson Soft Compilation, è una pubblicazione creata su misura per voi appassionati allo scopo di arricchire sempre di più la vostra raccolta di programmi.*

# SOMMARIO

# GOLF

Si sa! Il GOLF è il gioco dell'equilibrio per eccellenza; lo sport in cui il controllo delle braccia e la stabilità delle gambe giocano un ruolo importantissimo, tanto quanto la calma e la concentrazione che precedono l'esecuzione di ogni colpo.

Se da un lato il computer ci impigrisce, dall'altro riesce a coinvolgerci tanto che alla fine ci sembra di esserci stancati come se avessimo fatto veramente dello sport. Così ci sorprendiamo a prepararci prima di saltare un ostacolo, a concentrare i nostri sforzi al momento del salto e a riposarci alla fine di una corsa.
In questo GOLF l'abilità e la strategia necessarie nella realtà sono riprodotte verosimilmente e, come dal vero, è assolutamente necessario un buon allenamento prima

```
3 CALL CLEAR
5 PRINT TAB(8);"COPYRIGHT 1984"
7 PRINT "     VIRGIN GAMES LIMITED"::::
10 PRINT TAB(5);"*******************"
15 PRINT TAB(5);"*                 *"
20 PRINT TAB(5);"*      GOLF       *"
30 PRINT TAB(5);"*                 *"
40 PRINT TAB(5);"*       by        *"
50 PRINT TAB(5);"*                 *"
60 PRINT TAB(5);"*   P J FINKILL   *"
70 PRINT TAB(5);"*                 *"
80 PRINT TAB(5);"*******************"
90 PRINT :::::::
93 FOR DEL=1 TO 1000
95 NEXT DEL
100 CALL CLEAR
110 RANDOMIZE
120 CALL CHAR(128,"1C1E101010383838")
130 CALL CHAR(129,"0")
140 CALL CHAR(136,"0")
150 CALL CHAR(144,"183C7E7E3C181818")
160 CALL CHAR(152,"187E7EFFFF7E7E18")
170 CALL CHAR(120,"0000001818")
180 CALL CHAR(121,"FFFFFFFFFFFFFFFF")
190 CALL CHAR(112,"0842080080044108")
200 HOLE=1
210 DIM H(18,4),TSC(4),AVG(4),BL(4)
220 FOR I=1 TO 4
230 TSC(I)=0
240 AVG(I)=0
250 BL(I)=0
```

di poter mandare in buca qualche palla.
Notevole è il numero di opzioni: da 1 a 4 giocatori, da 1 a 18 buche (per gli sportivi meno dinamici).
Scelto il numero delle buche e dei giocatori, compare una legenda dei simboli usati per indicare le diverse zone del campo da gioco: il verde chiaro per indicare il prato, l'azzurro per la acqua, il verde scuro per gli alberi, un tondino nero per la buca ed il giallino per la sabbia.
La pallina lanciata corre sul prato, frena la sua corsa nella sabbia ma si perde nell'acqua, tra gli alberi e fuori dal campo da gioco.
A questo punto occorre dedicare particolare attenzione alle diverse possibilità di tiro a disposizione.
Esistono tre opzioni di scelta: innanzitutto, occorre scegliere la direzione di tiro, nord (E), nord est (R), est (D), sud est (C), sud (X), sud ovest (Z), ovest (S), nord ovest (W).
La seconda opzione riguarda il tipo di mazza, da 0 a 9; 9 = più pesante, 2 = più leggera, 1 = putter (tiro ravvicinato) e 0 = mazza da sabbia.
Infine, l'ultima scelta riguarda la forza del colpo: 1 = debole (ad esempio putter), 2 = colpo medio e 3 = colpo forte.
A questo proposito, se si colpisce la pallina con la massima forza c'è la possibilità di "tagliare" la palla.
Un aiuto: nel caso in cui ci

```
260 FOR J=1 TO 18
270 H(J,I)=0
280 NEXT J
290 NEXT I
300 INPUT "NUMBER OF PLAYERS(1 TO 4):":TPL

310 IF TPL<1 THEN 300
320 IF TPL>4 THEN 300
330 INPUT "NUMBER OF HOLES(1 TO 18)":TH
340 IF TH<1 THEN 330
350 IF TH>18 THEN 330
355 GOSUB 960
360 GOTO 3500
370 CALL CLEAR
380 CALL SCREEN(3)
390 SC=0
400 FOR I=11 TO 16
410 CALL COLOR(I,1,1)
420 NEXT I
430 PR$="HOLE "&STR$(HOLE)&" PLAYER "&STR$
(PL)
440 Z=3
450 GOSUB 3360
460 REM  trees
470 TC=INT(RND*5)+10
480 TR=INT(RND*2)+1
490 TN=INT(RND*10)+1
500 FOR I=TC TO TC+TN
510 N=INT(RND*6)+1
520 CALL VCHAR(TR,I,144,N)
530 NEXT I
540 TC=INT(RND*5)+10
550 TR=INT(RND*2)+11
560 TN=INT(RND*10)+1
570 FOR I=TC TO TC+TN
580 N=INT(RND*6)+1
590 CALL VCHAR(TR,I,144,N)
600 NEXT I
610 REM       water
620 TC=INT(RND*12)+10
630 TR=INT(RND*13)+1
640 TN=INT(RND*7)+1
650 FOR I=TC TO TC+TN
660 N=INT(RND*5)+1
670 CALL VCHAR(TR,I,136,N)
680 NEXT I
690 REM        bunker
700 N=INT(RND*4)+1
710 FOR I=1 TO N
720 TC=INT(RND*14)+16
730 TR=INT(RND*15)+2
740 CALL VCHAR(TR,TC,112,3)
750 CALL VCHAR(TR,TC+1,112,3)
760 NEXT I
770 REM       hole
```

si trovi ad una casella di distanza dalla buca scegliete la direzione corrispondente, date valore 1 al tipo di mazza (putter) ed ancora valore 1 alla forza del colpo (tiro debole); centrerete la buca perfettamente.
Dopo la legenda relativa ai diversi tipi di terreno, premuto il tasto, compaiono altre istruzioni: "dopo che avete deciso la forza con cui colpire la palla premete la barra spazio per colpire la palla" eccetera. Ah!... Dimenticavamo; premete il tasto alpha lock.

```
780 TR=INT(RND*12)+5
790 TC=INT(RND*12)+16
800 CALL VCHAR(TR-1,TC-3,129,3)
810 CALL VCHAR(TR-2,TC-2,129,5)
820 CALL VCHAR(TR-3,TC-1,129,7)
830 CALL VCHAR(TR-3,TC,129,7)
840 CALL VCHAR(TR-3,TC+1,129,7)
850 CALL VCHAR(TR-2,TC+2,129,5)
860 CALL VCHAR(TR-1,TC+3,129,3)
870 CALL VCHAR(TR,TC,128)
880 REM        tee
890 CALL HCHAR(11,2,121)
900 CALL HCHAR(11,3,120)
910 REM        bushes
920 CALL HCHAR(1,1,152,32)
930 CALL HCHAR(21,1,152,32)
940 CALL VCHAR(2,1,152,19)
950 CALL VCHAR(2,32,152,19)
953 GOSUB 960
955 GOTO 1030
960 REM      set colours
970 CALL COLOR(11,11,12)
980 CALL COLOR(12,16,1)
990 CALL COLOR(13,2,4)
1000 CALL COLOR(14,5,5)
1010 CALL COLOR(15,13,1)
1020 CALL COLOR(16,2,1)
1025 RETURN
1030 REM      set variables
1040 X1=11
1050 Y1=3
1060 RS1=32
1070 DIST=(TC-3)*10
1080 CALL HCHAR(24,1,32,32)
1090 PR$="DISTANCE="&STR$(DIST)&" METRES"

1100 Z=1
1110 GOSUB 3360
1120 FOR DELAY=1 TO 600
1130 NEXT DELAY
1140 CALL HCHAR(24,1,32,32)
1150 GOTO 2680
1160 REM   pick club/strength
1170 PR$="CLUB"
1180 Z=2
1190 GOSUB 3360
1200 CALL KEY(0,K,S)
1210 IF S=0 THEN 1200
1220 C=K-48
1230 CALL SOUND(100,250,10)
1240 IF C<10 THEN 1270
1250 CALL HCHAR(24,1,32,32)
1260 GOTO 1170
1270 PR$="STRENGTH"
1280 Z=10
```

La schermata successiva indica i tasti per le direzioni di tiro (già indicati) e le opzioni di scelta (indicate anche queste). A questo punto, premuto il tasto "Y" comparirà il variopinto campo da gioco. All'estrema sinistra si troverà la pallina bianca accanto ad un quadratino dello stesso colore e intorno alla buca da centrare appariranno combinazioni più o meno complicate di acqua, sabbia, alberi e prato, attraverso le quali bisogna passare per giungere all'obiettivo buca.

```
1290 GOSUB 3360
1300 CALL KEY(0,K,S)
1310 IF S=0 THEN 1300
1320 ST=K-48
1330 IF ST<1 THEN 1270
1340 IF ST>3 THEN 1270
1350 CALL SOUND(100,300,10)
1360 CALL KEY(0,K,S)
1370 IF S=0 THEN 1360
1380 IF K=32 THEN 1400
1390 GOTO 1360
1400 CALL SOUND(100,350,10)
1410 SC=SC+1
1420 IF RS1<>112 THEN 1500
1430 GOTO 1450
1440 IF RS1<>144 THEN 1500
1450 IF C=0 THEN 1480
1460 CALL SOUND(100,110,10)
1470 GOTO 2680
1480 C=2
1490 GOTO 1550
1500 IF RS1<>129 THEN 1550
1510 Q=7
1520 IF C<>1 THEN 1560
1530 LNH=C*ST+INT(RND)
1540 GOTO 1650
1550 Q=ST*2
1560 LNH=(ST*C)-5+(RND*Q)
1570 IF ST<>3 THEN 1650
1580 FLAG=INT(RND*3)
1590 IF FLAG<2 THEN 1630
1600 IF RR<>0 THEN 1630
1610 IF CC<>1 THEN 1630
1620 RR=1
1630 EX=INT(RND*4)
1640 LNH=LNH+EX
1650 REM      move ball
1660 FOR I=1 TO LNH
1670 X=X1+RR
1680 Y=Y1+CC
1690 CALL GCHAR(X,Y,RS)
1700 IF RS=152 THEN 1720
1710 GOTO 1740
1720 CALL HCHAR(X1,Y1,RS1)
1730 GOTO 1880
1740 CALL HCHAR(X1,Y1,RS1)
1750 CALL HCHAR(X,Y,120)
1760 X1=X
1770 Y1=Y
1780 RS1=RS
1790 NEXT I
1800 IF RS=144 THEN 1830
1810 IF RS=136 THEN 1830
1820 GOTO 1850
1830 CALL HCHAR(X,Y,RS)
```

In basso sullo schermo comparirà una scritta che indica la distanza che intercorre fra la pallina e la buca; sempre in basso, verranno via via formulate richieste riguardo la direzione, il tipo di mazza e la forma del colpo, accompagnate da suoni di accettazione per ogni risposta data.
Le prime volte sarà molto facile perdere le palline da gioco: "You've lost the ball", ma con molta pazienza e allenamento vi assicuro che sarà avvincente cercare di imbucarle nel minor numero di tiri possibile.

```
1840 GOTO 1880
1850 IF RS=128 THEN 2020
1860 CALL HCHAR(24,1,32,32)
1870 GOTO 2680
1880 REM    lost ball
1890 SC=SC+1
1900 CALL HCHAR(24,1,32,32)
1910 PR$="YOU'VE LOST THE BALL !!"
1920 Z=3
1930 GOSUB 3360
1940 CALL SOUND(100,392,15)
1950 CALL SOUND(100,330,15)
1960 CALL HCHAR(X1,Y1,120)
1970 BL(PL)=BL(PL)+1
1980 FOR DELAY=1 TO 1000
1990 NEXT DELAY
2000 CALL HCHAR(24,1,32,32)
2010 GOTO 2680
2020 REM    ball holed
2030 H(HOLE,PL)=SC
2040 CALL HCHAR(X1,Y1,RS1)
2050 CALL HCHAR(24,1,32,32)
2060 CALL SOUND(100,-6,10,1000,0)
2070 CALL SOUND(500,-6,10,1500,0)
2080 CALL SOUND(1500,-6,10,2000,0)
2090 PR$="YOU'VE HOLED IN "&STR$(SC)
2100 Z=3
2105 RS=32
2110 GOSUB 3360
2120 FOR DELAY=1 TO 1000
2130 NEXT DELAY
2140 IF PL=TPL THEN 2230
2150 PL=PL+1
2160 SC=0
2170 PR$="HOLE "&STR$(HOLE)&" PLAYER "&STR
$(PL)&"    "
2180 Z=2
2190 GOSUB 3360
2200 FOR DELAY=1 TO 1000
2210 NEXT DELAY
2220 GOTO 880
2230 IF HOLE=TH THEN 2270
2240 HOLE=HOLE+1
2250 PL=1
2260 GOTO 370
2270 REM       end
2280 FOR DELAY=1 TO 1000
2290 NEXT DELAY
2300 CALL CLEAR
2310 GOSUB 2330
2320 GOTO 2380
2330 PRINT TAB(10);"SCORECARD"
2340 PRINT TAB(9);"***********"
2350 PRINT TAB(14);"PLAYER NUMBER"
2360 PRINT TAB(12);"(1)";TAB(16);"(2)";TAB
```

Quando si riesce a centrare la buca compare una scritta che indica in quanti tiri si è riuscito a farlo e, successivamente, il campo da gioco della seconda buca.
Quando sono terminate tutte le buche comparirà uno schermo riassuntivo che indicherà per ogni buca e per ogni giocatore il numero dei tiri occorsi per centrare il bersaglio e, alla pressione di un tasto, il punteggio totale, il punteggio medio (per ogni buca) ed il numero delle palline perdute durante il gioco... da rimborsare, naturalmente!

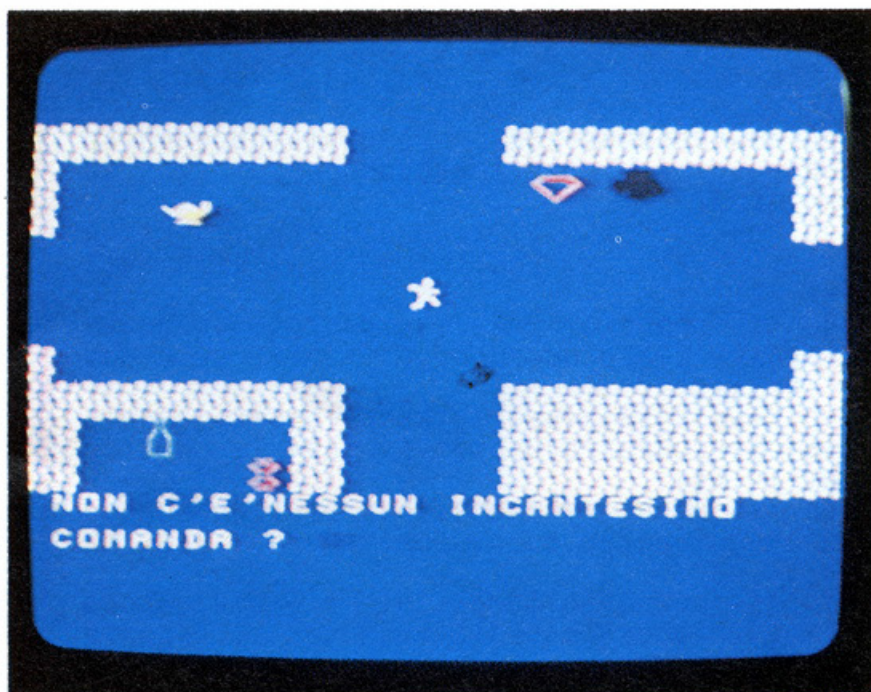
```
(20);"(3)";TAB(24);"(4)"
2370 RETURN
2380 FOR I=1 TO TH
2390 PRINT TAB(2);"HOLE ";I;TAB(12);H(I,1)
;TAB(16);H(I,2);TAB(20);H(I,3);TAB(24);H(I
,4)
2400 FOR J=1 TO 4
2410 TSC(J)=TSC(J)+H(I,J)
2420 NEXT J
2430 AD=0
2440 NEXT I
2450 PRINT
2460 PRINT "HIT ANY KEY TO CONTINUE"
2470 CALL KEY(0,K,S)
2480 IF S=0 THEN 2470
2490 CALL CLEAR
2500 GOSUB 2330
2510 FOR I=1 TO 4
2520 AVG(I)=INT(TSC(I)/TH)
2530 NEXT I
2540 PRINT
2550 PRINT "TOTAL "
2560 PRINT "SCORE";TAB(11);TSC(1);TAB(15);
TSC(2);TAB(19);TSC(3);TAB(23);TSC(4)::
2570 PRINT "AVERAGE"
2580 PRINT "SCORE";TAB(11);AVG(1);TAB(15);
AVG(2);TAB(19);AVG(3);TAB(23);AVG(4)::
2590 PRINT "BALLS"
2600 PRINT " LOST";TAB(11);BL(1);TAB(15);B
L(2);TAB(19);BL(3);TAB(23);BL(4)::
2610 PRINT ::::
2620 PRINT "PRESS 'Y' TO PLAY AGAIN"
2630 CALL KEY(0,K,S)
2640 IF S=0 THEN 2630
2650 IF K=89 THEN 2670
2660 GOTO 2630
2670 GOTO 100
2680 REM  pick direction
2690 CALL HCHAR(24,1,32,32)
2700 PR$="DIRECTION ? "
2710 Z=3
2720 GOSUB 3360
2730 CALL KEY(0,K,S)
2740 IF S=0 THEN 2730
2750 CALL SOUND(100,200,10)
2760 IF K<>69 THEN 2800
2770 RR=-1
2780 CC=0
2790 GOTO 3080
2800 IF K<>88 THEN 2840
2810 RR=1
2820 CC=0
2830 GOTO 3080
2840 IF K<>68 THEN 2880
2850 RR=0
2860 CC=1
```

```
2870 GOTO 3080
2880 IF K<>83 THEN 2920
2890 RR=0
2900 CC=-1
2910 GOTO 3080
2920 IF K<>82 THEN 2960
2930 RR=-1
2940 CC=1
2950 GOTO 3080
2960 IF K<>87 THEN 3000
2970 RR=-1
2980 CC=-1
2990 GOTO 3080
3000 IF K<>90 THEN 3040
3010 RR=1
3020 CC=-1
3030 GOTO 3080
3040 IF K<>67 THEN 2730
3050 RR=1
3060 CC=1
3070 REM
3080 CALL HCHAR(24,1,32,32)
3090 GOTO 1160
3100 REM    instructions
3110 CALL CLEAR
3120 PL=1
3130 CALL SCREEN(6)
3140 PRINT TAB(10);"***********"
3150 PRINT TAB(10);"*   GOLF  *"
3160 PRINT TAB(10);"***********"
3170 PRINT TAB(10);"DIRECTIONS"::
3180 PRINT TAB(10);" W E R"
3190 PRINT TAB(10);" S + D"
3200 PRINT TAB(10);" Z X C"::
3210 PRINT "CLUBS :"::
3220 PRINT "0 TO 9"
3230 PRINT "9 [HEAVIEST]-2 [LIGHTEST]"
3240 PRINT "1 = PUTTER"
3250 PRINT "0 = SAND WEDGE"::
3260 PRINT "STRENGTH OF STRIKE :"::
3270 PRINT "1 = SOFT(i.e.PUT)"
3280 PRINT "2 = MEDIUM STRIKE"
3290 PRINT "3 = HARDEST STRIKE"::
3300 PRINT "PRESS 'Y' TO START"
3310 CALL KEY(0,K,S)
3320 IF S=0 THEN 3310
3330 IF K=89 THEN 3350
3340 GOTO 3310
3350 GOTO 370
3360 REM   print routine
3370 FOR P=1 TO LEN(PR$)
3380 CALL HCHAR(24,Z+P,ASC(SEG$(PR$,P,1)))

3390 NEXT P
3400 RETURN
3500 CALL CLEAR
```

```
3505 CALL SCREEN(6)
3510 PRINT TAB(7);"****************"
3520 PRINT TAB(7);"*     GOLF      *"
3530 PRINT TAB(7);"*      BY       *"
3540 PRINT TAB(7);"* P J PROGRAMS *"
3550 PRINT TAB(7);"****************"::
3560 PRINT CHR$(129);"= THE GREEN"::
3570 PRINT CHR$(136);"= WATER"::
3580 PRINT CHR$(144);"= TREES"::
3590 PRINT CHR$(152);"= BUSHES"::
3600 PRINT CHR$(112);"= BUNKER"::
3610 PRINT "IF YOU HIT THE BALL AT "
3620 PRINT "MAXIMUM STRENGTH (IE 3) THEN"

3630 PRINT "THERE IS A CHANCE THAT THE"
3640 PRINT "BALL MAY BE SLICED."
3650 PRINT ::
3660 PRINT "HIT ANY KEY TO CONTINUE"
3670 CALL KEY(0,K,S)
3680 IF S=0 THEN 3670
3690 CALL CLEAR
3700 PRINT "AFTER YOU VE DECIDED ON THE"
3710 PRINT "STRENGTH TO HIT THE BALL "
3720 PRINT "THEN PRESS THE SPACE BAR TO"
3730 PRINT "HIT THE BALL."
3740 PRINT ::
3750 PRINT "IF THE BALL FAILS TO MOVE"
3760 PRINT "THEN YOU HAVE MISSED IT WITH"
3770 PRINT "YOUR SWING !"
3780 PRINT :::
3785 PRINT "PLEASE ENSURE ALPHA LOCK KEYIS
 ON (IE DOWN)"::::
3790 PRINT "HIT ANY KEY TO CONTINUE"
3800 CALL KEY(0,K,S)
3810 IF S=0 THEN 3800
3820 GOTO 3100
```

# Guida per l'input dei programmi sul TI 99/4A

A causa del sistema di codifica utilizzato, possono capitare in alcune linee delle improvvise "andate a capo" delle quali, naturalmente, non va tenuto conto in fase di digitazione.
Vi potrà capitare, a volte, di trovare nei listati di programmi per TI 99/4A alcuni caratteri sottolineati. La sottolineatura rappresenta una particolare codifica dei tasti di controllo: dovrete quindi premere il tasto indicato dal carattere insieme al tasto CONTROL.
Ad esempio, se trovate A dovrete battere CONTROL+A; se trovate –, dovrete battere CONTROL+SPAZIO, e così via.

# FUGA DAL LABIRINTO INCANTATO (Ex. Bas)

```
100 REM ******************
110 REM *     FUGA DAL     *
120 REM *     LABIRINTO    *
125 REM *     INCANTATO    *
130 REM ******************
140 REM
150 REM        TI 99/4A
160 REM    EXTENDED BASIC
170 RANDOMIZE
180 CALL CLEAR :: CALL SCREEN(5)
190 FOR I=1 TO 14 :: CALL COLOR(I,16,1)::
NEXT I
200 DISPLAY AT(9,9):"FUGA   DAL " :: DISPL
AY AT(11,5):"LABIRINTO INCANTATO"
210 CALL CHAR(136,"70777707E0EEEE0E"):: CA
LL HCHAR(2,3,136,28):: CALL HCHAR(23,3,136
,28)
220 CALL VCHAR(2,3,136,22):: CALL VCHAR(2,
30,136,22)
230 FOR DELAY=1 TO 500 :: NEXT DELAY :: CA
LL CLEAR
```

Il mago Torlik ti ha racchiuso in una buia prigione che non mostra alcuna apparente via d'uscita. Tenti disperatamente di dirigerti a nord e ti ritrovi apparentemente al punto di partenza. Tutte le stanze sembrano uguali... Ma ecco che appoggiandoti ad una pietra ti accorgi che dietro sono nascoste alcune strane cose...
Ed è qui che comincia la vostra avventura con questo misterioso gioco in Extended Basic per il vostro TI 99/4a.
Dopo il caricamento da cassetta vi troverete immersi

```
240 DIM N(21),S(21),E(21),OV(21)
250 FOR A=1 TO 21 :: READ N(A),S(A),E(A),O
V(A):: NEXT A
260 I=0 :: FOR A=1 TO 21 :: I=I+N(A)+S(A)+
E(A)+OV(A):: NEXT A :: READ CC :: IF I<>CC
 THEN PRINT "ISTRUZIONI-DATA-":"ERRATE " :
: STOP
270 R=1
275 GOSUB 2010 :: CALL CLEAR
280 CALL MAGNIFY(3)
290 A$="70777707E0EEEE0E"
300 CALL CHAR(116,A$):: CALL CHAR(124,A$):
: CALL CHAR(132,A$):: CALL CHAR(140,A$)
310 CALL COLOR(12,16,1)
320 A$="000038E838C80D0F070301010000000000
00000040E0F1F9FDFF680800000000" :: CALL CH
AR(136,A$)
330 FOR A=1 TO 5 :: READ NOUN$(A):: READ N
C(A):: NEXT A :: FOR I=1 TO 6 :: READ RH(I
):: NEXT I
340 A$="00000107CF7F3F1F07030F000000000000
000ECAEEFCF8F0E0C0F0000000000" :: CALL CHA
R(108,A$)
350 A$="1C3663410F1F787E5F591D1C161330001C
3663C0F8FC8FBFFD4D5C1C34E40700" :: CALL CH
AR(112,A$)
360 A$="00000207070203 1F03030F0818000000000
000000070D0C080808080E02000" :: CALL CH
AR(44,A$)
370 A$="000000000000E0B0301010101010704 0000
0040E0E040C0F8C0C0F0101800000 0" :: CALL CH
AR(40,A$)
380 A$="0001173F0F0FEF7FFFFFFF7B33270F0300
00B0E0F080D4FCF8FCFFFEFCF480C0" :: CALL CH
AR(100,A$)
390 A$="00000000001F3F70E070381C0E06030100
00000000F8FC1E070E1C387060C080" :: CALL CH
AR(96,A$)
400 A$="0007010F1C0E070202070E1C0F01070000
E080F03870E04040E07038F080E000" :: CALL CH
AR(92,A$)
410 A$="030101010103020604040404040404 07E0
4040404060203010101010101010F0" :: CALL CH
AR(104,A$)
420 A$="001C764207FF919D9F88CB0F1B11300000
386E42E0FF89B9F911D3F0D8880C00" :: CALL CH
AR(120,A$)
430 A$="0D121120211B050F0E040A691060106080
038C13A0C0A0F07020508804040A0A" :: CALL CH
AR(128,A$)
440 CALL COLOR(14,7,12)
450 FOR A=1 TO 5 :: O(A)=22 :: NEXT A
460 CALL CLEAR
470 FOR I=9 TO 14 :: CALL COLOR(I,1,1):: N
EXT I
```

in un'avventura in cui solo la vostra immaginazione saprà salvarvi.
Sin dall'inizio sarete in possesso di cinque oggetti misteriosi: una lampada, un'amuleto, una gemma, una pietra ed una bottiglia; lasciando uno di questi oggetti nelle varie stanze in cui avrete messo piede, riu-

```
480 FOR I=2 TO 8 :: CALL COLOR(I,16,1):: N
EXT I
490 CALL COLOR(11,16,1)
500 CALL SCREEN(5)
510 GOSUB 1040
520 GOSUB 820
530 GOTO 1350
540 REM USCITA NORD
550 CALL MOTION(#1,-20,0)
560 CALL COINC(#1,#17,16,V):: IF V=0 THEN
560 ELSE CALL COLOR(#7,1,#8,1,#9,1,#10,1,#
11,1,#12,1,#13,1,#14,1,#15,1)
570 CALL LOCATE(#1,128,121):: CALL COLOR(1
1,16,1,12,16,1,13,1,1,14,16,1)
580 CALL COINC(#1,#17,16,V):: IF V=0 THEN
580 ELSE CALL LOCATE(#1,128,121):: CALL CO
LOR(11,16,1,12,1,1,13,1,1,14,1,1)
590 CALL COINC(#1,#20,8,V):: IF V=0 THEN 5
90 ELSE CALL MOTION(#1,0,0):: CALL LOCATE(
#1,65,121)
600 RETURN
610 REM USCITA SUD
620 CALL MOTION(#1,20,0)
630 CALL COINC(#1,#18,16,V):: IF V=0 THEN
630 ELSE CALL COLOR(#7,1,#8,1,#9,1,#10,1,#
11,1,#12,1,#13,1,#14,1,#15,1)
640 CALL LOCATE(#1,1,121):: CALL COLOR(11,
16,1,12,16,1,13,1,1,14,16,1)
650 CALL COINC(#1,#18,16,V):: IF V=0 THEN
650 ELSE CALL LOCATE(#1,1,121):: CALL COLO
R(11,16,1,12,1,1,13,1,1,14,1,1)
660 CALL COINC(#1,#20,8,V):: IF V=0 THEN 6
60 ELSE CALL MOTION(#1,0,0):: CALL LOCATE(
#1,65,121)
670 RETURN
680 REM USCITA EST
690 CALL MOTION(#1,0,20)
700 CALL COINC(#1,#19,16,V):: IF V=0 THEN
700 ELSE CALL COLOR(#7,1,#8,1,#9,1,#10,1,#
11,1,#12,1,#13,1,#14,1,#15,1)
710 CALL LOCATE(#1,65,1):: CALL COLOR(11,1
6,1,12,1,1,13,16,1,14,16,1)
720 CALL COINC(#1,#19,16,V):: IF V=0 THEN
720 ELSE CALL LOCATE(#1,65,1):: CALL COLOR
(11,16,1,12,1,1,13,1,1,14,1,1)
730 CALL COINC(#1,#20,8,V):: IF V=0 THEN 7
30 ELSE CALL MOTION(#1,0,0):: CALL LOCATE(
#1,65,121)
740 RETURN
750 REM USCITA OVEST
760 CALL MOTION(#1,0,-20)
770 CALL COINC(#1,#16,16,V):: IF V=0 THEN
770 ELSE CALL COLOR(#7,1,#8,1,#9,1,#10,1,#
11,1,#12,1,#13,1,#14,1,#15,1)
780 CALL LOCATE(#1,65,241):: CALL COLOR(11
```

scirete ad orientarvi verso la via d'uscita. Per fare qualsiasi azione servitevi delle parole che il computer capisce ed... attenzione! Una delle stanze possiede un fascino magico e così anche uno dei cinque oggetti. Se vi troverete nella stanza incantata con l'oggetto incantato e pronuncerete la parola ma-

gica... vi troverete finalmente liberi... Ma se mancherete di uno di questi ingredienti vi troverete trasportati in una stanza normale, magari in compagnia di una delle creature che popolano il labirinto... (a proposito, non sempre dovete dare ascolto a quanto vi dicono... potrebbe costarvi caro e trovarvi

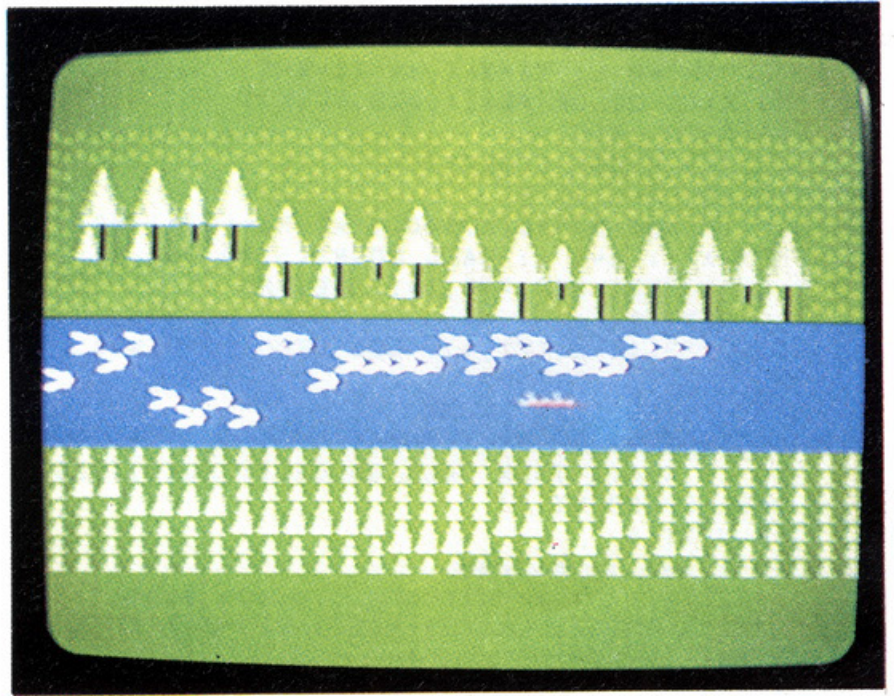
```
,16,1,12,1,1,13,16,1,14,16,1)
790 CALL COINC(#1,#16,16,V):: IF V=0 THEN
790 ELSE CALL LOCATE(#1,65,241):: CALL COL
OR(11,16,1,12,1,1,13,1,1,14,1,1)
800 CALL COINC(#1,#20,8,V):: IF V=0 THEN 8
00 ELSE CALL MOTION(#1,0,0):: CALL LOCATE(
#1,65,121)
810 RETURN
820 REM STANZA
830 CALL COLOR(11,16,1)
840 CALL COLOR(12,1,1)
850 CALL COLOR(13,1,1)
860 CALL COLOR(14,1,1)
870 GOSUB 1960
880 RETURN
890 REM TRAPPOLE
900 IF R<>RH(6)THEN 950
910 CALL SCREEN(7)
920 CALL COLOR(#1,2)
930 DISPLAY AT(21,7):" F O R N A C E " ::
DISPLAY AT(22,6):"PARTITA CONCLUSA" :: DIS
PLAY AT(23,3):"SEI STATO CARBONIZZATO"
940 CALL SOUND(4250,-6,1):: GOTO 1020
950 CALL COLOR(11,16,1)
960 CALL COLOR(12,1,1)
980 CALL COLOR(14,1,1)
990 FOR I=7 TO 12 :: CALL HCHAR(I,1,116,2)
:: CALL HCHAR(I,31,116,2):: NEXT I
1000 FOR I=14 TO 19 :: CALL VCHAR(1,I,116,
2):: CALL VCHAR(15,I,116,6):: NEXT I
1010 DISPLAY AT(22,5):"PARTITA CONCLUSA:SE
I" :: DISPLAY AT(23,3):"MURATO VIVO NELLA
CAMERA"
1020 CALL SOUND(1000,110,0):: CALL SOUND(2
00,123,0):: CALL SOUND(600,131,0):: CALL S
OUND(2000,110,0)
1030 GOTO 2030
1040 CC=1 :: REM  ROUTINE CREAZIONE SPRITE
S
1050 CALL SPRITE(#1,40,16,65,121,0,0)
1060 CALL SPRITE(#2,96,NC(1),145,25,0,0,#7
,96,CC,17,161,0,0)
1070 CALL SPRITE(#3,100,NC(2),145,49,0,0,#
8,100,CC,17,185,0,0)
1080 CALL SPRITE(#4,92,NC(3),145,73,0,0,#9
,92,CC,17,217,0,0)
1090 CALL SPRITE(#6,104,NC(5),129,41,0,0,#
11,104,CC,33,25,0,0)
1100 CALL SPRITE(#5,108,NC(4),129,65,0,0,#
10,108,CC,33,49,0,0)
1110 CALL SPRITE(#12,112,CC,49,33,0,0)
1120 CALL SPRITE(#13,120,CC,49,193,0,0)
1130 CALL SPRITE(#14,128,CC,89,65,0,0)
1140 CALL SPRITE(#15,136,CC,89,177,0,0)
1150 CALL SPRITE(#16,88,CC,65,1,0,0)
1160 CALL SPRITE(#17,88,CC,1,121,0,0)
```

```
1170 CALL SPRITE(#18,88,CC,137,121,0,0)
1180 CALL SPRITE(#19,88,CC,65,241,0,0)
1190 CALL SPRITE(#20,88,CC,65,121,0,0)
1200 CALL HCHAR(1,1,116,640)
1210 FOR I=17 TO 20 :: CALL HCHAR(I,4,32,8
):: NEXT I
1220 FOR I=3 TO 14 :: CALL HCHAR(I,3,140,2
8):: NEXT I
1230 FOR I=7 TO 12 :: CALL HCHAR(I,1,124,3
2):: NEXT I
1240 FOR I=1 TO 20 :: CALL HCHAR(I,14,132,
6):: NEXT I
1250 FOR I=7 TO 12 :: CALL HCHAR(I,14,32,6
):: NEXT I
1260 RETURN
1270 REM STANZA MAGICA
1280 CALL SOUND(200,-5,0):: CALL COLOR(11,
11,1):: DISPLAY AT(21,1):"SEI NELLA STANZA
 MAGICA"
1290 RETURN
1300 REM LIBERAZIONE
1310 CALL SOUND(200,-5,0):: CALL COLOR(11,
1,1):: CALL SCREEN(13):: DISPLAY AT(3,7):"
CONGRATULAZIONI"
1320 DISPLAY AT(12,7):"SEI RIUSCITO AD" ::
 DISPLAY AT(14,8):"E V A D E R E"
1330 CALL SOUND(100,196,1):: CALL SOUND(10
0,262,1):: CALL SOUND(100,392,1):: CALL SO
UND(300,523,1):: CALL SOUND(500,440,1)
1340 CALL SOUND(500,523,1):: GOTO 2040
1350 REM COMPRENSIONE FRASI
1360 NN$,VB$="" :: DISPLAY AT(23,1):"COMAN
DA ? " :: ACCEPT AT(23,11):A$
1370 FOR I=21 TO 24 :: CALL HCHAR(I,3,32,2
8):: NEXT I
1380 FOR A=1 TO LEN(A$)
1390 IF SEG$(A$,A,1)=" " THEN X=A-1 :: A=0
 :: GOTO 1420
1400 NEXT A
1410 VB$=A$ :: GOTO 1460
1420 VB$=SEG$(A$,1,X)
1430 NN$=SEG$(A$,X+2,LEN(A$)-X-1)
1440 REM COMPRENSIONE VERBI
1450 IF VB$="VAI" THEN VB$=NN$
1460 IF VB$="NORD" OR VB$="SUD" OR VB$="ES
T" OR VB$="OVEST" OR VB$="N" OR VB$="S" OR
 VB$="E" OR VB$="O" THEN 1560
1470 IF VB$="PRENDI" OR VB$="MUOVI" THEN 1
770
1480 IF VB$="POSA" OR VB$="LASCIA" THEN 18
30
1490 IF VB$="Q" OR VB$="QUIT" THEN CALL CL
EAR :: STOP
1500 IF VB$<>"DI'" THEN 1550
1510 IF R<>21 THEN DISPLAY AT(21,1):"NON C
```

murati vivi in una cella!). Per impartire gli ordini sollevate il tasto ALPHA LOCK e battete il comando: ad es. "Posa Pietra" (Non mettete gli articoli!). Se un ordine non viene compreso, viene risposto di riprovare...
Non c'è altro da aggiungere se non... Buona Fortuna!
*Tratto da "Paper Book"*

```
'E'NESSUN INCANTESIMO" :: GOTO 1360
1520 IF O(3)<>22 THEN DISPLAY AT(22,1):"MA
NCA L'OGGETTO INCANTATO"
1530 IF NN$<>"PIETRA" THEN DISPLAY AT(21,1
):"PAROLA MAGICA ERRATA"
1540 IF NN$="PIETRA" AND O(3)=22 THEN 1300
 ELSE R=1 :: GOSUB 2070 :: GOTO 1360
1550 DISPLAY AT(22,1):"NON CONOSCO :  ";VB
$ :: GOTO 1360
1560 REM MOVIMENTI
1570 CALL SOUND(50,1245,3)
1580 VV$=SEG$(VB$,1,1):: IF VV$="N" OR VV$
="O" THEN CALL PATTERN(#1,44)ELSE CALL PAT
TERN(#1,40)
1590 IF VB$="N" OR VB$="NORD" THEN X,R=N(R
):: GOSUB 540
1600 IF VB$="S" OR VB$="SUD" THEN X,R=S(R)
:: GOSUB 610
1610 IF VB$="E" OR VB$="EST" THEN X,R=E(R)
:: GOSUB 680
1620 IF VB$="O" OR VB$="OVEST" THEN X,R=OV
(R):: GOSUB 750
1630 I=INT(RND*4)+1 :: IF I=1 THEN A$="EST
" ELSE IF I=2 THEN A$="OVEST" ELSE IF I=3
THEN A$="SUD" ELSE A$="NORD"
1640 IF R=RH(1)THEN CALL COLOR(#12,10):: D
ISPLAY AT(21,1):"UNA BESTIA INFUOCATA DICE
" :: DISPLAY AT(22,1):"VA'A ";A$
1650 IF R=RH(2)THEN CALL COLOR(#13,2):: DI
SPLAY AT(21,1):"UNO SCIMMIONE DICE" :: DIS
PLAY AT(22,1):"VA'A ";A$
1660 IF R=RH(3)THEN CALL COLOR(#15,12):: D
ISPLAY AT(21,1):"UN DRAGO DICE" :: DISPLAY
 AT(22,1):"LA FORNACE E' A ";A$
1670 IF R=RH(4)THEN CALL COLOR(#14,16):: D
ISPLAY AT(21,1):"UN RAGNO GIGANTE DICE" ::
 DISPLAY AT(22,1):"NON ANDARE A ";A$
1680 IF R=RH(5)THEN 890
1690 IF R=RH(6)THEN 890
1700 IF R=21 THEN GOSUB 1270 :: GOTO 1720

1710 GOSUB 820
1720 X=R
1730 GOSUB 1960
1740 X=0
1750 CALL SOUND(50,988,3):: CALL SOUND(50,
988,3)
1760 GOTO 1360
1770 REM ROUTINE PRENDERE
1780 GOSUB 1900
1790 IF O(X)=R OR O(X)=22 THEN 1810
1800 GOTO 1820
1810 O(X)=22 :: XA=X+1 :: XB=X+6 :: CALL C
OLOR(#XA,NC(X),#XB,1)
1820 X,XA,XB=0 :: GOTO 1360
```

```
1830 REM ROUTINE LASCIARE
1840 PX=0 :: GOSUB 1900 :: IF PX=1 THEN PX
=0 :: GOTO 1360
1850 IF O(X)=22 THEN 1870
1860 GOTO 1890
1870 O(X)=R :: XA=X+1 :: XB=X+6
1880 CALL COLOR(#XA,1,#XB,NC(X))
1890 GOTO 1360
1900 REM PRESA OGGETTI
1910 FOR A=1 TO 5
1920 IF NN$=NOUN$(A)AND(O(A)=R)+(O(A)=22)T
HEN X=A :: A=0 :: CALL SOUND(-200,660,1)::
 CALL SOUND(-200,440,1):: RETURN
1930 NEXT A
1940 DISPLAY AT(21,1):"NON VEDO LA ":NN$
1950 PX=1 :: RETURN
1960 REM ROUTINE ELENCO
1970 FOR A=1 TO 5 :: Y=A+6
1980 IF X=O(A)THEN CALL COLOR(#Y,NC(A))ELS
E CALL COLOR(#Y,1)
1990 NEXT A
2000 A,Y=0 :: RETURN
2010 PRINT "Le parole che capisco sono:":"
 ":"POSA-PRENDI-LASCIA-MUOVI-DI'-VAI-NORD-
SUD-EST-OVEST"
2020 PRINT "GEMMA-PIETRA-BOTTIGLIA-":"LAMP
ADA-AMULETO "
2025 FOR PAUSA=1 TO 3000 :: NEXT PAUSA ::
RETURN
2030 REM NUOVA PARTITA
2040 DISPLAY AT(24,5):"GIOCHI ANCORA? (S/N
)"
2050 CALL KEY(0,KY,ST):: IF KY<>83 AND KY<
>78 THEN 2050
2060 IF KY=78 THEN CALL CLEAR :: STOP ELSE
 R=1 :: CALL SCREEN(5):: GOTO 450
2070 CALL SOUND(500,165,1):: CALL SOUND(99
9,110,1):: CALL SCREEN(2)
2080 CALL COLOR(11,7,1,12,7,1,13,7,1,14,7,
1):: CALL SOUND(200,-5,1)
2090 CALL SCREEN(5):: CALL COLOR(11,16,1,1
2,1,1,13,1,1,14,1,1):: X=R :: GOSUB 1960 :
: RETURN
2100 REM DATA
2110 DATA 1,2,20,6,20,2,20,7,4,3,1,2,4,4,4
,4,15,21,15,20,6,20,6,20,7,8,7,3
2120 DATA 3,7,8,9,14,14,4,14,10,10,15,6,11
,6,16,11,7,12,11,12,8,13,18,13
2130 DATA 13,14,14,14,15,10,5,15,10,11,16,
16,17,12,17,17,19,17,19,19
2140 DATA 19,19,19,19,20,6,1,20,20,20,20,2
0,997
2150 DATA GEMMA,10,PIETRA,2,AMULETO,10,LAM
PADA,12,BOTTIGLIA,8,3,9,18,5,4,19
```

# CACCIA ALL'ORSO

Vi ha mai tentati la vita del Trapper? Sempre a contatto con la natura e con l'unica preoccupazione di riempire la pancia prima dello scadere del giorno. Vivere situazioni sempre nuove ogni giorno, tra pericoli come indiani, puma, serpenti, orsi... Un momento, fermiamoci qui.

Ora avete la possibilità di vivere simili situazioni alla ricerca di un orso, il terribile Grizzly, col quale placare la fame per qualche giorno.

Dopo aver scelto se vivere un'avventura impossibile, dura, o qualcosa di intermedio e dopo aver controllato che il tasto ALPHA LOCK sia sollevato, vi troverete tra foreste incontaminate a bordo della vostra canoa in compagnia del vostro fedele

```
100 REM    ********************
110 REM    * CACCIA ALL'ORSO *
120 REM    ********************
130 REM          JACKSON
140 REM    TI 99/4A BASIC
150 OPTION BASE 1
160 DIM A(31,11)
170 CALL CLEAR
180 RANDOMIZE
190 FOR I=1 TO 31
200 A(I,1)=10
210 A(I,11)=10
220 NEXT I
230 DATA 10,10,10,10,10,10,400,350,300,-30
0
240 READ A(1,10),A(2,10),A(3,10),A(29,10),
A(30,10),A(31,10),SN(2),SN(3),SN(4),SN(5)
250 RESTORE 3950
260 FOR X=1 TO 22
270 READ B,A$
280 CALL CHAR(B,A$)
290 NEXT X
300 FOR X=1 TO 5
310 READ D,B,C
320 CALL COLOR(D,B,C)
330 NEXT X
340 GOTO 3580
350 GOSUB 370
```

```
360 GOTO 1630
370 CALL CLEAR
380 CALL COLOR(9,3,13)
390 CALL HCHAR(1,1,96,320)
400 M=1
410 K=3
420 CALL COLOR(10,2,13)
430 GOSUB 1440
440 K=5
450 GOSUB 1440
460 K=6
470 GOSUB 1440
480 GOSUB 1440
490 CALL HCHAR(18,1,137,224)
500 CALL SCREEN(13)
510 FOR I=3 TO 27 STEP 2
520 CALL HCHAR(INT(RND*5+19),I,138,2)
530 NEXT I
540 F=14
550 R=-1
560 CALL HCHAR(11,1,150,32)
570 CALL HCHAR(17,1,149,32)
580 CALL HCHAR(12,1,130,160)
590 IF R=W THEN 1120
600 X=INT(RND*5)+2
610 R=R+1
620 D=1
630 J=2
640 T=R+2
650 Z=-1-R
660 U=1-R*.15
670 A(J,X)=T
680 A(J+1,X)=T
690 IF R>1 THEN 710
700 A(J+2,X)=T
710 ON INT(RND*1.55+1.75)GOTO 720,750,770

720 IF (X-D>6)+(X-D<2)THEN 770
730 D=-D
740 GOTO 760
750 IF (X+D>6)+(X+D<2)THEN 720
760 X=X+D
770 X4=INT(RND*5+2)
780 IF A(J,X4)=T THEN 770
790 A(J,X4)=Z
800 CALL HCHAR(10+X4,J,132)
810 IF RND>U THEN 770
820 J=J+1
830 IF J<27 THEN 670
840 I=31
850 CALL KEY(3,X,Y)
860 I=I-1
870 IF X=69 THEN 900
880 IF X=88 THEN 920
890 GOTO 930
900 F=F-1
```

Winchester, della vostra Colt e delle immancabili provviste pronti per affrontare le rapide...
Già dovete districarvi tra scogli più o meno vicini senza sbattervi contro. Ogni volta che urterete uno di essi perdete qualcosa tra ciò che tenete a bordo; nell'ordine se ne andranno: la prima volta il fucile (potrete sparare all'orso da vicino), la pistola (non potrete cacciare l'orso) per seconda, e se sbatterete anche una terza volta perderete le provviste e non avrete più scampo dalla morte per inedia. Dopo aver superato le tre serie di rapide, accostate a riva in prossimità della pista dell'orso e premete il tasto -O- (occhio a non scambiarlo per lo zero!) dopodichè,

```
910 GOTO 930
920 F=F+1
930 CALL HCHAR(F,I,115,2)
940 IF (F<12)+(F>16)THEN 1000
950 IF (A(I,F-10)=Z)+(A(I+1,F-10)=Z)THEN 1
010
960 IF I=1 THEN 580
970 CALL SOUND(SN(T),-5,20)
980 CALL HCHAR(F,I,130,2)
990 GOTO 850
1000 F=14
1010 CALL SOUND(1500,110,13,1000,15,-7,3)

1020 FOR K=I TO 1 STEP -1
1030 CALL GCHAR(14,K,G)
1040 CALL HCHAR(14,K,117)
1050 CALL SOUND(200,-5,20)
1060 CALL HCHAR(14,K,G)
1070 NEXT K
1080 A(30,7)=A(30,7)-7
1090 H=0
1100 IF A(30,7)<A(1,7)THEN 3500
1110 GOTO 560
1120 RETURN
1130 K=22
1140 M=2
1150 GOSUB 1360
1160 A$="GIOCHI ANCORA?(S/N)"
1170 K=24
1180 M=2
1190 GOSUB 1360
1200 CALL HCHAR(12,1,130,160)
1210 J=30
1220 R=0
1230 R=1-R
1240 CALL HCHAR(14,J,115,2)
1250 CALL KEY(3,I,K)
1260 IF I=78 THEN 4010
1270 IF I=83 THEN 3580
1280 IF R=0 THEN 1300
1290 CALL HCHAR(14,J,130,2)
1300 J=J-R
1310 IF J>1 THEN 1230
1320 J=30
1330 GOTO 1230
1340 FOR NL=1 TO L
1350 READ K,M,A$
1360 FOR I=1 TO LEN(A$)
1370 CALL HCHAR(K,M+I,ASC(SEG$(A$,I,1)))
1380 NEXT I
1390 IF NL=0 THEN 1430
1400 NEXT NL
1410 L=1
1420 NL=0
1430 RETURN
```

usando i tasti con le frecce come in precedenza, cercate l'orso. Quando finalmente sarete a portata di tiro premete la E (maiuscola), quando la musichetta vi avviserà che avete colpito l'orso prendetevelo e correte nella canoa (per entrare premete il tasto -I-) e passate alle rapide successive, dopodiché sarete di nuovo a

```
1440 I=0
1450 M=M+2
1460 CALL HCHAR(K,M,154)
1470 CALL HCHAR(K,M+1,155)
1480 CALL HCHAR(K+1,M,156)
1490 CALL HCHAR(K+1,M+1,157)
1500 CALL HCHAR(K+2,M,158)
1510 CALL HCHAR(K+2,M+1,159)
1520 CALL HCHAR(K+3,M,137)
1530 CALL HCHAR(K+4,M,138)
1540 CALL VCHAR(K+3,M+1,104,2)
1550 I=I+1
1560 ON I GOTO 1450,1570,1620
1570 M=M+1
1580 CALL HCHAR(K+1,M+1,152)
1590 CALL HCHAR(K+2,M+1,153)
1600 CALL VCHAR(K+3,M+1,105)
1610 GOTO 1450
1620 RETURN
1630 CALL HCHAR(1,1,137,320)
1640 CALL COLOR(9,4,12)
1650 CALL HCHAR(10,5,96,23)
1660 FOR I=4 TO 28 STEP 2
1670 IF I=28 THEN 1690
1680 CALL VCHAR(2,I+1,96,8)
1690 FOR J=2 TO 9
1700 A(I,J)=10
1710 NEXT J
1720 NEXT I
1730 FOR I=6 TO 24 STEP 6
1740 F=INT(RND*5)+4
1750 A(I,F)=0
1760 CALL HCHAR(F,I,96)
1770 G=INT(RND*(F-3))+2
1780 A(I,G)=0
1790 CALL HCHAR(G,I,96)
1800 A(I+1,G+1)=10
1810 CALL HCHAR(G+1,I+1,137)
1820 F=INT(RND*5+2)
1830 IF F=G+1 THEN 1820
1840 A(I+2,F)=0
1850 CALL HCHAR(F,I+2,96)
1860 G=INT(RND*(7-F))+F
1870 A(I+2,G)=0
1880 CALL HCHAR(G,I+2,96)
1890 A(I+3,G+1)=10
1900 CALL HCHAR(G+1,I+3,137)
1910 A(I+4,10)=10
1920 CALL HCHAR(10,I+4,137)
1930 A(I+4,9)=0
1940 CALL HCHAR(9,I+4,96)
1950 A(I+4,3)=0
1960 CALL HCHAR(3,I+4,96)
1970 NEXT I
1980 A(4,10)=10
```

casa. Attenzione a non perdere troppo tempo per la caccia all'orso, altrimenti potreste perdere la vostra stessa vita. E fate attenzione a non essere voi stessi la preda da cacciare: l'orso è molto pericoloso!!! In bocca all'...orso!

*Tratto da: Paper Soft N° 2/1984*

```
1990 CALL HCHAR(10,4,137)
2000 A(28,9)=10
2010 CALL HCHAR(9,28,137)
2020 A(28,3)=10
2030 CALL HCHAR(3,28,137)
2040 N=14
2050 O=30
2060 CALL HCHAR(N,O,115,2)
2070 G=INT(RND*4+1)*6-1
2080 CALL COLOR(10,2,11)
2090 F=INT(RND*5)+3
2100 D=1
2110 CALL HCHAR(24,3,116,A(30,7)-3)
2120 CALL HCHAR(22,5,84)
2130 CALL HCHAR(22,6,69)
2140 CALL HCHAR(22,7,77)
2150 CALL HCHAR(22,8,80)
2160 CALL HCHAR(22,9,79)
2170 C=9
2180 CALL KEY(3,I,J)
2190 CALL HCHAR(N,O,130,2)
2200 IF I=83 THEN 2240
2210 IF I=69 THEN 2290
2220 IF I=88 THEN 2320
2230 IF I=79 THEN 2420 ELSE 2260
2240 O=O-1
2250 IF O=1 THEN 3270
2260 CALL HCHAR(N,O,115,2)
2270 GOSUB 2350
2280 GOTO 2180
2290 IF N=12 THEN 2270
2300 N=N-1
2310 GOTO 2260
2320 IF N=16 THEN 2270
2330 N=N+1
2340 GOTO 2260
2350 C=C-1
2360 IF C>-1 THEN 2410
2370 A(30,7)=A(30,7)-1
2380 C=A(30,8)
2390 CALL HCHAR(24,A(30,7),137)
2400 IF A(30,7)=3 THEN 3450
2410 RETURN
2420 IF N>12 THEN 2260
2430 IF A(O,10)=10 THEN 2260
2440 CALL HCHAR(N,O,115,2)
2450 P=N-2
2460 X6=O
2470 J=1
2480 CALL HCHAR(P,X6,96)
2490 CALL KEY(5,I,K)
2500 IF I=101 THEN 2670
2510 IF I=120 THEN 2680
2520 IF I=115 THEN 2710
2530 IF I=100 THEN 2720
```

```
2540 IF I=105 THEN 2910
2550 IF I=83 THEN 2830
2560 IF I=69 THEN 2750
2570 IF I=68 THEN 2840
2580 IF I=88 THEN 2760
2590 CALL HCHAR(P,X6,107)
2600 GOSUB 2350
2610 IF H<=1 THEN 2920
2620 IF (F<>P)+(G<>X6)THEN 2470
2630 H=2000
2640 CALL SOUND(800,220,5,330,6,440,7)
2650 F=0
2660 GOTO 2470
2670 J=-1
2680 IF A(X6,P+J)=10 THEN 2590
2690 P=P+J
2700 GOTO 2590
2710 J=-1
2720 IF A(X6+J,P)=10 THEN 2590
2730 X6=X6+J
2740 GOTO 2590
2750 J=-1
2760 CALL SOUND(5,110,1,-3,1)
2770 FOR I=P+J TO P+Q*J STEP J
2780 IF (I<>F)+(X6<>G)THEN 2810
2790 CALL SOUND(1000,300,3)
2800 H=H+1
2810 NEXT I
2820 GOTO 2590
2830 J=-1
2840 CALL SOUND(5,110,1,-3,1)
2850 FOR I=X6+J TO X6+Q*J STEP J
2860 IF (I<>G)+(P<>X1)THEN 2890
2870 CALL SOUND(1000,330,3)
2880 H=H+1
2890 NEXT I
2900 GOTO 2590
2910 IF (P=10)*(X6=0)+(P=10)*(X6=0+1)THEN
2180 ELSE 2590
2920 CALL HCHAR(F,G,96)
2930 IF (F=P)*(G=X6)THEN 3550
2940 V(1)=0
2950 V(2)=0
2960 V(3)=0
2970 V(4)=0
2980 IF A(G,F+1)=10 THEN 3000
2990 V(1)=1
3000 IF A(G+1,F)=10 THEN 3020
3010 V(2)=1
3020 IF A(G,F-1)=10 THEN 3040
3030 V(3)=1
3040 IF A(G-1,F)=10 THEN 3060
3050 V(4)=1
3060 B=D-2
3070 IF B>0 THEN 3090
3080 B=B+4
```

```
3090 ON V(1)+V(2)+V(3)+V(4)GOTO 3100,3120,
3130,3130
3100 D=B
3110 GOTO 3160
3120 IF (V(1)+V(3)=0)+(V(2)+V(4)=0)THEN 31
60
3130 D=INT(RND*4+1)
3140 IF D=B THEN 3130
3150 IF V(D)=0 THEN 3130
3160 K=-1
3170 ON D GOTO 3180,3210,3190,3220
3180 K=1
3190 F=F+K
3200 GOTO 3230
3210 K=1
3220 G=G+K
3230 IF ABS(F-P)+ABS(G-X6)>A(1,8)THEN 3250

3240 CALL HCHAR(F,G,106)
3250 IF (F=P)*(G=X6)THEN 3550
3260 GOTO 2470
3270 A(1,7)=A(1,7)+6
3280 FOR I=2 TO 27
3290 FOR J=2 TO 6
3300 A(I,J)=0
3310 NEXT J
3320 NEXT I
3330 M=4
3340 CALL COLOR(9,3,13)
3350 CALL COLOR(10,2,13)
3360 GOSUB 440
3370 IF H>1999 THEN 3430
3380 IF A(30,7)>6 THEN 3410
3390 A$="L'HAI SCAMPATA BELLA!"
3400 GOTO 1130
3410 A$="CE L'HAI FATTA!!!"
3420 GOTO 1130
3430 A$=" GRANDE CACCIA!!!"
3440 GOTO 1130
3450 A$="SEI MORTO PER LA FAME!!"
3460 FOR I=1 TO 10
3470 CALL SOUND(-1000,1200-100*I,2*I)
3480 NEXT I
3490 GOTO 1130
3500 A$="SEI ANNEGATO!"
3510 FOR I=1 TO 20 STEP 2
3520 CALL SOUND(-100,110,I,-2,I)
3530 NEXT I
3540 GOTO 1130
3550 A$="SBRANATO DALL'ORSO!"
3560 CALL SOUND(2000,760,10,-3,15)
3570 GOTO 1130
3580 A(30,7)=25
3590 W=2
3600 A(30,8)=16
```

```
3610 Q=4
3620 A(1,8)=30
3630 A(1,7)=-6
3640 CALL CLEAR
3650 CALL SCREEN(4)
3660 H=1
3670 CALL COLOR(10,7,1)
3680 FOR I=2 TO 10
3690 FOR J=3 TO 27
3700 A(J,I)=0
3710 NEXT J
3720 CALL HCHAR(6,2*I+4,106)
3730 NEXT I
3740 L=7
3750 RESTORE 3930
3760 GOSUB 1340
3770 CALL HCHAR(24,31,46)
3780 CALL KEY(3,S,E)
3790 IF S=49 THEN 350
3800 IF S=50 THEN 3820
3810 IF S=51 THEN 3870 ELSE 3780
3820 A(30,8)=12
3830 A(1,8)=8
3840 A(1,7)=1
3850 W=3
3860 GOTO 350
3870 A(30,8)=8
3880 A(1,8)=4
3890 Q=2
3900 A(1,7)=7
3910 W=3
3920 GOTO 350
3930 DATA 4,7,CACCIA  ALL' ORSO,12,4,VUOI
UNA PARTITA,14,4,1. DURA,16,4,2. DIFFICILE
,18,4,3. IMPOSSIBILE
3940 DATA 22,4,SOLLEVA IL TASTO ALPHA LOCK
,24,4,COSA SCEGLI?
3950 DATA 130,0,132,E07A1E07071E7AE0,115,1
0301099FFFF,116,FFFFFFFFFFFFFFFF,117,00002
0D30C03
3960 DATA 159,FAF0FAF0FAFEF8FF,158,1F033F0
31F3F1F7F,157,E080F0C0F0C0F8E0,156,0701070
10F030F03
3970 DATA 155,8080C080C080E080,154,0000010
0030003,153,7E187E1CFE1CFF1C,152,08081C081
E083E18
3980 DATA 96,40E044A40E,137,0008081C083C18
3E,138,183E3C7E3CFE7E,150,FFFF9F1E0A80,104
,8080808080808080
3990 DATA 105,0808080808080808,106,4438381
0FE386C6C,107,00081C2A0A181414,149,0000001
03078F9FF
4000 DATA 13,16,5,11,9,5,16,4,13,14,4,13,1
5,5,5
4010 END
```

# TEXT EDITOR

L'uso del personal come elaboratore di testi è ormai così diffuso che è diventata una tendenza comune, soprattutto nel campo dei portatili, corredare il computer di un Text Editor residente in ROM, eventualmente accompagnato da altri programmi di data base, wailing list e grafica commerciale.
Il TI99/4A dispone, a tale proposito, del modulo SSS Writer TI, forse difficilmente reperibile in commercio al giorno d'oggi. Il programma comprende molte funzioni avanzate le quali tuttavia non compensano, a nostro avviso, la necessità di dover leggere il testo di 80 colonne attraverso finestre contigue ed in parte sovrapposte di 40 caratteri ciascuna. Come giustamente è scritto in *Things to do with your TI99/4A computer*, J. Willis, M. Miller & D. La Mont Johnson, New American Library, dopo aver usato il Writer TI vi sembrerà di aver assistito ad una partita di tennis da bordo campo!
Molti usano l'Editor/Assembler come word-processor, ma è chiaro che oltre a non eliminare il problema delle finestre, esso comporta la perdita di molte importanti funzioni come l'inserimento di una parola con il riaggiustamento dell'intero capoverso. In altre parole un inserimento è possibile solo nell'ambito di una linea di testo e comporta la perdita della parte finale della linea stessa. Inoltre con l'opzione Print dell'Editor/Assembler non è possibile controllare il formato della stampa.
Dopo questa premessa non sembrerà più tanto infondata la proposta di un nuovo word-processor per il TI99/4A.
Scritto in Extended BASIC, il programma si attiene alle seguenti caratteristiche di base: normale output su video con 28 caratteri per riga, con la possibilità di usare un nuovo set di caratteri stile IBM, con un vero corsivo; testo in forma di capoversi; controllo del formato di stampa tramite codici inseriti tra un capoverso e l'altro; giustificazione del margine destro di stampa con suddivisione in sillabe per il ritorno a capo riga (!); mantenimento di tutte le principali funzioni dell'Editor/Assembler, quali Delete, Move Copy, Insert, che permettono la cancellazione e lo spostamento di grossi blocchi di testo compreso il merge e la fusione di due testi diversi.
Il programma occupa circa 12 Kbyte di memoria e necessita del sistema a dischi, l'espansione di memoria e la stampante. Esso permette di gestire un testo fino ad un massimo di 300 righe per 28 caratteri, corrispondente al 15 videate.
La carta di identità di questo text-editor è abbastanza eloquente e permette di collocarlo su un piano di competitività, se non di superiorità, rispetto ad altri programmi analoghi quali *Direct Writer II* della Dynamic Data & Devices, *Word-Processing* della Futura Software Inc. o *Textiger*, tutti reperibili negli U.S.A. Il programma è diviso in due blocchi: il primo denominato DSK1. Textnewset, contiene la definizione del nuovo set di caratteri; il secondo DSK1. Texteditor, è il programma vero e proprio. Il primo si allaccia automaticamente al secondo, il quale può anche essere eseguito in modo autonomo, nel qual caso però si dispone solo del normale set di caratteri.
Il menù fornisce 6 opzioni, nell'ordine: Load, Edit, Save, Purge e Files. Con Load si carica in memoria un file precedentemente registrato. Edit permette di creare un testo o editare quello presente in memoria. Parleremo più avanti di tutte le funzioni di editing. Save memorizza il testo presente in memoria nel formato Sequential, Display, Variable 80. Ogni record contiene due righe di testo di 28 caratteri ed il file così ottenuto può

```
100 REM    NUOVO SET
110 REM   DI CARATTERI
120 REM ****************
130 CALL CLEAR :: CALL CHAR(33,"0018181818
180018"):: CALL CHAR(34,"006C6C6C"):: CALL
 CHAR(37,"0073566C18366ACE")
140 !@P-
150 CALL CHAR(40,"0008103030301008"):: CAL
L CHAR(41,"0010080C0C0C0810"):: CALL CHAR(
43,"000018187E7E1818")
155 CALL CHAR(45,"000000007E7E0000"):: CAL
L CHAR(38,"002050502OD6CC76")
160 CALL CHAR(47,"0002060C18306040"):: CAL
L CHAR(48,"003C66666E76663C"):: CALL CHAR(
49,"001838181818183C")
170 CALL CHAR(50,"003C66060C18307E"):: CAL
L CHAR(51,"003C66060C06663C"):: CALL CHAR(
52,"000C1C2C4C7E0C0C")
180 CALL CHAR(53,"007E607C0606663C"):: CAL
L CHAR(54,"003C60607C66663C"):: CALL CHAR(
55,"007E460C18306060")
190 CALL CHAR(56,"003C66663C66663C"):: CAL
L CHAR(57,"003C66663E06663C"):: CALL CHAR(
60,"000C18306030180C")
200 CALL CHAR(62,"0030180C060C1830"):: CAL
L CHAR(63,"003C66660C180018"):: CALL CHAR(
65,"00183C66667E6666")
210 CALL CHAR(66,"00FC66667C6666FC"):: CAL
L CHAR(67,"003C66606060663C"):: CALL CHAR(
68,"00FC6666666666FC")
220 CALL CHAR(69,"00FE6260786062FE"):: CAL
L CHAR(70,"00FE6260786060F0"):: CALL CHAR(
71,"003C66606E66663C")
230 CALL CHAR(72,"006666667E666666"):: CAL
L CHAR(73,"003C18181818183C"):: CALL CHAR(
74,"003C181818185830")
240 CALL CHAR(75,"0062666C78786C66"):: CAL
L CHAR(76,"00F06060606062FE"):: CALL CHAR(
77,"00C6EEFED6C6C6C6")
250 CALL CHAR(78,"00C6E6F6DECEC6C6"):: CAL
L CHAR(79,"003C66666666663C"):: CALL CHAR(
80,"00FC6666667C60F0")
260 CALL CHAR(81,"003C6666666A643A"):: CAL
L CHAR(82,"00FC66667C6666F3"):: CALL CHAR(
83,"003C66603C06663C")
270 CALL CHAR(84,"007E5A181818183C"):: CAL
LCHAR(85,"006666666666663C"):: CALL CHAR(
86,"00666666243C1818")
280 CALL CHAR(87,"00C6D6D6D6D67C28"):: CAL
L CHAR(88,"0066663C183C6666"):: CALL CHAR(
89,"0066663C18181818")
290 CALL CHAR(90,"007E460C1830627E"):: CAL
L CHAR(91,"003C30303030303C"):: CALL CHAR(
93,"003C0C0C0C0C0C3C")
300 CALL CHAR(97,"000000380C7CCC76"):: CAL
L CHAR(98,"0060607C6666667C"):: CALL CHAR(
99,"0000003C6660663C")
310 CALL CHAR(100,"0006063E6666663E"):: CA
LL CHAR(101,"0000003C667E603C"):: CALL CHA
R(102,"001C303078303030")
320 CALL CHAR(103,"0000003A663E463C"):: CA
```

essere caricato anche con l'Editor Assembler. Sia per il comando Load che per il Save, va specificato il nome del file secondo le norme consuete: ad esempio è valido il nome DSK1. Myfile. Il comando Print permette la stampa del testo attualmente in memoria. È necessario specificare il nome del dispositivo, ad esempio RS-232 e successivamente il margine sinistro di stampa e la larghezza del testo. I valori di default sono 10 e 60, nel qual modo vengono lasciate 10 colonne di margine a destra e a sinistra del foglio. Ulteriori codici di controllo della stampa possono venir inseriti nel testo così da ottenere la stampa ingrandita, condensata, enfatizzata, doppia ed inoltre il salto riga o l'avanzamento del modulo, ecc. Il comando Purge annulla completamente il testo in memoria, dopo la conferma da parte dell'utente. Infine, con Files si ottiene la lista dei file Dis/Var 80 con la possibilità di cancellarli, uno alla volta, dalla directory.

Veniamo ora all'editing. Selezionato il comando Edit. si passa allo schermo contenente il testo. Questo occupa le 20 righe centrali. Le prime due righe, nello schermo 1, contengono il nome del file, o la scritta Newfile se si inizia un nuovo testo; negli schermi successivi, nella stessa posizione, vengono mostrate le ultime due righe della videata precedente, in

modo da consentire continuità nella scrittura. Le ultime due righe in basso sono riservate alle funzioni Delete, Move, Copy, ecc.
Durante la scrittura si hanno a disposizione due cursori, uno "vuoto" ed uno "pieno" lampeggiante, corrispondenti a due diversi modo di editing. Si passa da un modo all'altro semplicemente premendo il tasto Enter. Il cursore lampeggiante indica che ci si trova nell'editing di linea; si può scrivere, spostare il cursore a destra e a sinistra, inserire e cancellare caratteri, ma solo nell'ambito di una riga. Premendo Enter compare il cursore vuoto a capo riga ed in questo modo si ha a disposizione un editing di schermo, potendo spostare il cursore in ogni parte del video, cancellare o inserire righe, inserire o cancellare caratteri con il riaggiustamento dell'intero capoverso, accedere alle funzioni Move, Copy, Insert, ecc.
Durante la scritta è necessario premere il tasto Enter al termine di ogni riga. Se si desidera usufruire delle funzioni generali di editing basta un leggero tocco e si ottiene il cursore vuoto a capo riga, se invece si desidera continuare la scrittura basta mantenere la pressione sul tasto Enter e si ottiene il cursore "pieno" lampeggiante, sempre a capo riga. Per scrivere bisogna sempre essere in presenza del cursore lampeggiante.

```
LL CHAR(104,"0060607C66666666"):: CALL CHA
R(105,"001800381818183C")
330 CALL CHAR(106,"0018003818185830"):: CA
LL CHAR(107,"006060666C786C66"):: CALL CHA
R(108,"003818181818183C")
340 CALL CHAR(109,"000000ECB6B6B6B6"):: CA
LL CHAR(110,"0000005C66666666"):: CALL CHA
R(111,"0000003C6666663C")
350 CALL CHAR(112,"0000007C66667C60"):: CA
LL CHAR(113,"0000003E66663E06"):: CALL CHA
R(114,"0000006E76606060")
360 CALL CHAR(115,"0000003C603C067C"):: CA
LL CHAR(116,"0030307C3030301C"):: CALL CHA
R(117,"000000666666663C")
370 CALL CHAR(118,"0000006666663C18"):: CA
LL CHAR(119,"000000C6D6D67C28"):: CALL CHA
R(120,"000000663C183C66")
380 CALL CHAR(121,"000000663C185830"):: CA
LL CHAR(122,"0000007E4C18327E"):: CALL CHA
R(123,"001C30306030301C")
390 CALL CHAR(125,"00380C0C060C0C38"):: CA
LL CHAR(128,"FF818181818181FF")
400 RUN "DSK1.TEXTEDITOR"
100 REM  TEXT EDITOR 28/C
110 REM ------------------
120 REM by Sergio Borsani.
130 ON BREAK NEXT :: ON WARNING NEXT
140 DIM RIGA$(300):: NUMRIGHE=1 :: F$="
    *** (EOF) ***          " :: RIGA$(1)=F$
:: FINA$="NEW"
150 CALL CLEAR :: CALL SCREEN(13):: FOR J=
1 TO 12 :: CALL COLOR(J,16,13):: NEXT J ::
 CALL COLOR(14,16,1)
160 CALL CHAR(128,"FF818181818181FF"):: CA
LL CHAR(129,"FF"):: CALL CHAR(130,"0000000
0000000FF")
170 CALL CHAR(131,RPT$("01",8)):: CALL CHA
R(132,RPT$("80",8)):: CALL CHAR(136,"3C429
9A1A199423C"):: GOTO 250
180 CALL HCHAR :: CALL VCHAR :: SC$ :: N :
: RECORD$ :: X :: SCHERMO :: FINE :: CALL
SPRITE :: RI :: CO
190 R$ :: NU :: CALL KEY :: K :: S :: CALL
 DELSPRITE :: CALL LOCATE :: Z$ :: KK :: W
$ :: F1 :: T
200 R1 :: R2 :: R3 :: BLOCCO :: CTR :: Q :
: PX :: FNW$ :: FINA$ :: MS :: LF :: UR ::
 PNT :: RIG$ :: DEV$
210 PX1 :: PX2 :: RIGA0$ :: LR0 :: ST$ ::
STS :: L1$ :: L2$ :: L3$ :: L4$
220 CALL VOCALE :: CALL MARGINE :: V :: DI
FF :: COD :: LOOP :: FN$ :: W
230 FLAG :: DN$ :: Y :: PY
240 !@P-
250 CALL CLEAR :: DISPLAY AT(2,4):"T E X T
    E D I T O R" :: DISPLAY AT(4,2):"copyr
ight ";CHR$(136);"1984, S.Borsani"
260 CALL HCHAR(1,3,129,28):: CALL HCHAR(5,
3,130,28):: CALL VCHAR(1,2,131,5):: CALL V
CHAR(1,31,132,5)
270 DISPLAY AT(8,4):"1  LOAD" :: DISPLAY A
```

```
T(10,4):"2  EDIT" :: DISPLAY AT(12,4):"3
SAVE"
280 DISPLAY AT(14,4):"4  PRINT" :: DISPLAY
 AT(16,4):"5  PURGE"
290 DISPLAY AT(18,4):"6  FILES" :: DISPLAY
 AT(20,10):"scelta?"
300 CALL VCHAR(7,4,32,12):: ON BREAK STOP
:: ACCEPT AT(20,18)SIZE(1)VALIDATE("123456
"):SC$
310 ON ERROR 2490 :: ON BREAK NEXT :: N=VA
L(SC$):: CALL HCHAR(6+2*N,4,62):: ON N GOT
O 320,390,1270,1310,2020,2040
320 GOSUB 2200 :: ON ERROR 2220
330 NUMRIGHE=1 :: OPEN #1:FINA$,SEQUENTIAL
,INPUT ,DISPLAY ,VARIABLE 80
340 IF EOF(1)THEN 380
350 LINPUT #1:RECORD$ :: X=POS(RECORD$,F$,
1)
360 FOR J=0 TO 1 :: RIGA$(NUMRIGHE)=SEG$(R
ECORD$,28*J+1,28):: IF X=28*J+1 THEN 380
370 NUMRIGHE=NUMRIGHE+1 :: NEXT J :: GOTO
340
380 CLOSE #1 :: GOSUB 2390 :: GOTO 300
390 CALL CLEAR :: SCHERMO=1 :: IF NUMRIGHE
>20 THEN FINE=20 ELSE FINE=NUMRIGHE
400 GOSUB 2410 :: FOR J=1 TO FINE :: DISPL
AY AT(J+2,1):RIGA$(J):: NEXT J
410 CALL SPRITE(#1,128,2,17,17)
420 RI=3 :: CO=1 :: GOTO 480
430 GOSUB 2250 :: ACCEPT AT(RI,CO)SIZE(-(2
9-CO)):R$ :: R$=R$&RPT$(" ",29-CO-LEN(R$))

440 IF CO=1 THEN RIGA$(NU)=R$ ELSE RIGA$(N
U)=SEG$(RIGA$(NU),1,CO-1)&R$
450 CO=1 :: IF RI<22 THEN RI=RI+1 :: GOTO
470
460 RI=3 :: SCHERMO=SCHERMO+1 :: CALL CLEA
R :: GOSUB 2330 :: GOSUB 2320 :: IF NU=NUM
RIGHE THEN DISPLAY AT(3,1):F$
470 CALL SPRITE(#1,128,2,(RI-1)*8+1,17)
480 CALL KEY(5,K,S):: IF S=0 THEN 480
490 IF K=13 THEN CALL DELSPRITE(#1):: GOTO
 430
500 IF K=1 THEN 630
510 IF K=3 THEN GOSUB 2320 :: Y=CO :: GOTO
 790
520 IF K=4 THEN 720
530 IF K=6 THEN 780
540 IF K=7 THEN 690
550 IF K=8 AND CO>1 THEN CO=CO-1 :: GOTO 6
20
560 IF K=9 AND CO<28 THEN CO=CO+1 :: GOTO
620
570 IF K=10 AND RI<22 THEN RI=RI+1 :: GOSU
B 2250 :: GOTO 620
580 IF K=11 AND RI>3 THEN RI=RI-1 :: GOTO
620
590 IF K=12 AND SCHERMO>1 THEN 820
600 IF K=14 THEN 830
610 IF K=15 THEN 840 ELSE 480
620 CALL LOCATE(#1,(RI-1)*8+1,(CO+1)*8+1):
: GOTO 480
```

Anche se si deve premere Enter al termine di ogni riga, ogni capoverso o l'intero testo, sono virtualmente considerati dal programma come un'unica lunga riga. Pertanto, quando si va a capo non si devono rispettare le comuni regole grammaticali per la suddivisione in sillabe, né usare i trattini di unione. Ad esempio, se si sta scrivendo la parola computer e si termina la riga con "comp", si continuerà a capo con il seguito "uter". Analogamente, terminando una riga con una parola completa, si porrà uno spazio all'inizio della riga successiva, esattamente come se si proseguisse sulla stessa riga. Ci penserà poi il programma a formattare il testo nel modo specificato dall'utente.
Il test può essere suddiviso in capoversi di lunghezza variabile. Al limite tutto il testo può essere costituito da un unico capoverso, ma in tal caso l'esecuzione di alcune funzioni risulterebbe più lenta. I capoversi devono venir separati da una riga vuota con una barra obliqua inversa nella prima colonna (/ = Fctn z).
Dopo aver ripetuto con il cursore nero lampeggiante si dispone delle funzioni di editing di linea, cioè Delete Char, Insert Char, Erase, e dei comandi di controllo cursore a destra e a sinistra, ma solo nell'ambito di una riga, vediamo le funzioni generali di editing.

A queste si accede in presenza del cursore vuoto, il quale compare premendo il tasto Enter. Sono abilitati i seguenti tati di funzione: Delete Char = Fctn 1; Insert Char = Fctn 2; Delete Line = Fctn 3; Next Screen = Fctn 5; Roll Down = Fctn 6: Aid = Fctn 7; Insert Line = Fctn 8 ed Escape = Fctn 9. Attenzione perché, mentre è disabilitato il comando Break, tranne che in presenza del menu, è sempre attivo il comando Quit = Fctn + che corrisponde ad un reset di sistema.

Il significato delle funzioni è evidente, tuttavia è opportuno fare qualche precisazione. Del Char e Ins Char hanno valore per l'intero capoverso. Delete Line cancella la riga dove è posto il cursore e riaggiusta il testo. Next Screen, mancando le finestre di testo laterali, ha lo stesso significato del comando Roll Up dell'Editor/Assembler, cioè fa vedere lo schermo successivo. Roll Down, invece, riporta allo schermo precedente. Aid permette di ottenere la numerazione delle righe ed il numero dello schermo attuale. Insert Line inserisce una linea vuota in corrispondenza del cursore che può essere spostato in ogni posizione del video usando i tasti con le frecce. Infine. Escape mette a disposizione ulteriori comandi generali di editing che appaiono indicati nelle righe inferiori del video. Essi sono nell'ordine:

```
630 FOR J=3 TO 11 :: CALL HCHAR(J,2,46+J):
: NEXT J
640 FOR J=13 TO 21 :: CALL HCHAR(J,2,36+J)
:: NEXT J
650 Z$="SCHERMO "&STR$(SCHERMO)&"  RIGHE "
&STR$((SCHERMO-1)*20+1)&"-"&STR$(SCHERMO*2
0)
660 DISPLAY AT(24,1):Z$
670 CALL KEY(0,KK,S):: IF S=0 THEN 670
680 CALL VCHAR(3,2,32,20):: CALL HCHAR(24,
3,32,28):: GOTO 480
690 GOSUB 2320 :: IF NU=NUMRIGHE OR NUMRIG
HE=1 THEN 480
700 FOR J=NU TO NUMRIGHE-1 :: RIGA$(J)=RIG
A$(J+1):: NEXT J :: NUMRIGHE=NUMRIGHE-1 ::
 GOSUB 2330
710 CALL HCHAR(J,3,32,28):: GOTO 480
720 GOSUB 2320 :: W$=SEG$(RIGA$(NU),28,1):
: IF CO=1 THEN RIGA$(NU)=CHR$(32)&SEG$(RIG
A$(NU),1,27):: GOTO 740
730 RIGA$(NU)=SEG$(RIGA$(NU),1,CO-1)&CHR$(
32)&SEG$(RIGA$(NU),CO,28-CO)
740 NU=NU+1 :: IF NU=NUMRIGHE OR SEG$(RIGA
$(NU),1,1)="\" THEN 750 ELSE 770
750 IF W$<>CHR$(32)THEN GOSUB 2370 :: RIGA
$(NU)=W$&RPT$(" ",27)
760 GOSUB 2330 :: GOTO 480
770 Z$=SEG$(RIGA$(NU),28,1):: RIGA$(NU)=W$
&SEG$(RIGA$(NU),1,27):: W$=Z$ :: GOTO 740

780 GOSUB 2320 :: GOSUB 2370 :: GOSUB 2330
 :: GOTO 480
790 IF NU+1=NUMRIGHE OR SEG$(RIGA$(NU+1),1
,1)="\" THEN W$=CHR$(32):: F1=1 ELSE W$=SE
G$(RIGA$(NU+1),1,1)
800 IF Y=1 THEN RIGA$(NU)=SEG$(RIGA$(NU),2
,27)&W$ ELSE RIGA$(NU)=SEG$(RIGA$(NU),1,Y-
1)&SEG$(RIGA$(NU),Y+1,28-Y)&W$
810 IF F1=1 THEN F1=0 :: GOSUB 2330 :: GOT
O 480 ELSE NU=NU+1 :: Y=1 :: GOTO 790
820 SCHERMO=SCHERMO-1 :: CO=1 :: RI=3 :: C
ALL CLEAR :: GOSUB 2330 :: GOTO 480
830 IF SCHERMO*20>=NUMRIGHE THEN 480 ELSE
SCHERMO=SCHERMO+1 :: RI=3 :: CO=1 :: CALL
CLEAR :: GOSUB 2330 :: GOTO 620
840 CALL DELSPRITE(#1):: DISPLAY AT(23,1):
"  <E><M><C><D><I> <ESCAPE>":" " :: FOR T=
1 TO 50 :: NEXT T
850 CALL KEY(5,K,S):: IF S=0 THEN 850
860 IF K=15 THEN CALL DELSPRITE(#1):: GOTO
 250
870 IF K=69 OR K=101 THEN GOSUB 2390 :: GO
TO 620
880 IF K=77 OR K=109 THEN 920
890 IF K=67 OR K=99 THEN 1050
900 IF K=68 OR K=100 THEN 1110
910 IF K=73 OR K=105 THEN 1150 ELSE 850
920 DISPLAY AT(23,1):"MOVE) DA,A,DOPO LA R
IGA" :: ON ERROR 2470 :: ACCEPT AT(24,1):R
$ :: GOSUB 2500
930 IF R3<R1 OR R3>R2 AND R1<=R2 THEN 960
```

```
940 DISPLAY AT(24,1):"* ERRORE * PRESS <EN
TER>."
950 CALL KEY(0,K,S):: IF S=0 OR K<>13 THEN
 950 ELSE 840
960 BLOCCO=R2-R1+1 :: IF NUMRIGHE+BLOCCO<3
00 THEN 990
970 GOSUB 2390 :: DISPLAY AT(24,1):"* MEMO
RY FULL * PRESS ENTER."
980 CALL KEY(0,K,S):: IF S=0 OR K<>13 THEN
 980 ELSE 840
990 CTR=NUMRIGHE :: FOR J=R1 TO R2 :: CTR=
CTR+1 :: RIGA$(CTR)=RIGA$(J):: NEXT J
1000 IF R3>R2 THEN 1020
1010 FOR J=R2 TO R3+BLOCCO+1 STEP -1 :: RI
GA$(J)=RIGA$(J-BLOCCO):: NEXT J :: GOTO 10
30
1020 FOR J=R1 TO R3-BLOCCO :: RIGA$(J)=RIG
A$(J+BLOCCO):: NEXT J
1030 IF R3<R1 THEN Q=0 ELSE Q=BLOCCO
1040 FOR J=1 TO BLOCCO :: RIGA$(R3-Q+J)=RI
GA$(NUMRIGHE+J):: NEXT J :: GOTO 390
1050 DISPLAY AT(23,1):"COPY) DA,A,DOPO LA
RIGA" :: ON ERROR 2470 :: ACCEPT AT(24,1):
R$ :: GOSUB 2500
1060 IF R3<R1 OR R3>=R2 AND R1<=R2 THEN 10
70 ELSE 940
1070 BLOCCO=R2-R1+1 :: IF NUMRIGHE+BLOCCO>
300 THEN 970
1080 CTR=0 :: FOR J=NUMRIGHE+BLOCCO TO R3+
BLOCCO+1 STEP -1 :: RIGA$(J)=RIGA$(NUMRIGH
E-CTR):: CTR=CTR+1 :: NEXT J
1090 NUMRIGHE=NUMRIGHE+BLOCCO :: IF R3>R2
THEN Q=0
1100 CTR=0 :: FOR J=R3+1 TO R3+BLOCCO :: R
IGA$(J)=RIGA$(R1+Q+CTR):: CTR=CTR+1 :: NEX
T J :: GOTO 390
1110 DISPLAY AT(23,1):"DELETE) DA RIGA,A R
IGA" :: ON ERROR 2470 :: ACCEPT AT(24,1):R
$ :: GOSUB 2500
1120 IF R1>R2 OR R1<1 OR R2>NUMRIGHE-1 THE
N 940
1130 Q=R2-R1+1 :: CTR=0 :: FOR J=R1 TO NUM
RIGHE-Q :: CTR=CTR+1 :: RIGA$(J)=RIGA$(R2+
CTR):: NEXT J
1140 NUMRIGHE=NUMRIGHE-Q :: GOTO 390
1150 DISPLAY AT(23,1):"INSERT) FILE NAME,A
FTER LINE" :: ON ERROR 2470 :: DISPLAY AT(
24,1):"DSK1."
1160 ACCEPT AT(24,1)SIZE(-28):R$ :: PX=POS
(R$,",",1):: IF PX=0 OR PX>16 THEN 940
1170 FNW$=SEG$(R$,1,PX-1):: R3=VAL(SEG$(R$
,PX+1,LEN(R$)-PX)):: BLOCCO=0
1180 OPEN #1:FNW$,SEQUENTIAL,INPUT ,DISPLA
Y ,VARIABLE 80
1190 IF EOF(1)THEN 1210
1200 LINPUT #1:RECORD$ :: X=POS(RECORD$,F$
,1):: IF X=0 THEN BLOCCO=BLOCCO+2 :: GOTO
1190 ELSE BLOCCO=BLOCCO+INT(X/28)
1210 IF BLOCCO+NUMRIGHE>300 THEN CLOSE #1
:: GOTO 970 ELSE RESTORE #1 :: CTR=0
```

Edit, Move, Copy, Delete, Insert ed ancora Escape, tutte richiamabili con la sola lettera iniziale tranne Escape ottenibile nel solito modo con Fctn 9.

**Edit** permette di tornare nel modo edit.

**Move** consente di rimaneggiare il testo spostando un certo numero di righe in un'altra posizione. Dopo aver osservato i numeri di riga con il tasto Aid, si devono indicare i numeri della prima e dell'ultima linea del segmento da spostare ed il numero di riga oltre la quale va posto, tutti separati da virgole. (Move) 10,15,35 sposta la parte di testo dalla riga 10 alla 15 dopo la linea 35.

**Copy** svolge una funzione simile a Move, con la differenza che la parte di testo indicata non viene rimossa, ma solo copiata.

**Delete** permette la cancellazione di un blocco di testo specificando la prima e l'ultima riga della parte da cancellare.

**Insert** consente la fusione di due file, ovvero di quello attualmente in memoria con un altro memorizzato in precedenza su disco. È necessario specificare il nome del file del quale si richiede il merging ed il numero di riga dopo il quale dev'essere inserito. Con questo comando è possibile, ad esempio, richiamare l'intestazione di una lettera o un indirizzo tratto da una mailing list.

I comandi Copy e Insert pos-

sono portare ad una saturazione della memoria disponibie nel quale caso compare il messaggio ★ MEMORY FULL ★ PRESS ENTER, che rende invalido il comando impedendo la sua esecuzione senza che il programma debba subire un blocco.

**Escape** fa tornare al menu principale.

Consideriamo ora il modo in cui si possono inserire nel testo codici di controllo per la stampa. Si è detto che i capoversi in cui viene suddiviso il testo devono venir separati da una riga con una barra obliqua inversa (Fctn z) nella prima posizione. Ebbene, dopo tale segno si possono inserire i codici decimali, riportati nel manuale della stampante, ognuno seguito da una barra obliqua inversa. Ad esempio, /10/ produce il salto di una riga; /15/ causa la stampa di caratteri condensati; e ancora /27/69/ ha come effetto la stampa nel modo enfatizzato, ecc.

In pratica si possono usare tutti i codici previsti dal manuale; l'unica limitazione sta nel fatto che non è possibile inserire nella stessa riga diverse modalità di stampa. Tuttavia si può sempre ottenere l'intestazione di una lettera con caratteri diversi, ad esempio condensati, evidenziare un titolo con caratteri enfatizzati, ecc. Resta inteso che il programma riesce a tener allineato

```
1220 FOR J=NUMRIGHE+BLOCCO TO R3+BLOCCO+1
STEP -1 :: RIGA$(J)=RIGA$(J-BLOCCO):: NEXT
 J
1230 IF EOF(1)THEN 1260 ELSE LINPUT #1:REC
ORD$ :: X=POS(RECORD$,F$,1)
1240 FOR J=0 TO 1 :: IF X=28*J+1 THEN 1260
 ELSE CTR=CTR+1 :: RIGA$(R3+CTR)=SEG$(RECO
RD$,28*J+1,28)
1250 NEXT J :: GOTO 1230
1260 CLOSE #1 :: NUMRIGHE=NUMRIGHE+BLOCCO
:: GOTO 390
1270 GOSUB 2200 :: J=1 :: ON ERROR 2220
1280 OPEN #1:FINA$,SEQUENTIAL,OUTPUT,DISPL
AY ,VARIABLE 80
1290 RECORD$=RIGA$(J)&RIGA$(J+1):: PRINT #
1:RECORD$ :: IF J+1<NUMRIGHE THEN J=J+2 ::
 GOTO 1290
1300 CLOSE #1 :: GOSUB 2390 :: GOTO 300
1310 GOSUB 2190 :: IF SC$="N" OR SC$="n" T
HEN GOSUB 2390 :: GOTO 300
1320 GOSUB 2390 :: DISPLAY AT(23,1):"devic
e name:" :: DISPLAY AT(24,1):"RS232"
1330 ACCEPT AT(24,1)SIZE(-28):DEV$ :: CALL
 CLEAR
1340 ON ERROR 2430 :: OPEN #4:DEV$ :: DISP
LAY AT(18,1):"* STAMPA *":" ":"MARGINE SIN
ISTRO = 10"
1350 ACCEPT AT(20,20)SIZE(-2)VALIDATE(DIGI
T):MS :: IF MS<1 OR MS>80 THEN 1370
1360 DISPLAY AT(21,1):"NUMERO COLONNE = 60
" :: ACCEPT AT(21,18)SIZE(-2)VALIDATE(DIGI
T):LF :: IF LF>1 AND LF<80-MS+1 THEN 1390

1370 DISPLAY AT(23)BEEP:"* ERRORE * FORMAT
O STAMPA":"* PRESS <ENTER> TO CONTINUE"
1380 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 1380 ELSE GOSUB 2390 :: GOTO 1350
1390 DISPLAY AT(23,1):"PER INTERROMPERE LA
 STAMPA  PREMI (FCTN)(9)=ESCAPE."
1400 UR=NUMRIGHE-1 :: J=1 :: W=MS :: ON ER
ROR 2440
1410 PNT=LF :: RIG$=""
1420 CALL KEY(5,K,S):: IF K=15 THEN CLOSE
#4 :: GOTO 250
1430 IF SEG$(RIGA$(J),1,1)="\" THEN PX1=1
:: GOTO 1930
1440 RIGAO$=RIGA$(J):: LRO=LEN(RIGAO$)
1450 IF PNT<LRO THEN 1490
1460 RIG$=RIG$&RIGAO$ :: PNT=PNT-LRO :: J=
J+1
1470 IF PNT=0 THEN 1500
1480 IF J>UR THEN 1500 ELSE 1420
1490 RIG$=RIG$&SEG$(RIGAO$,1,PNT)
1500 ST$=RIG$ :: STS=0
1510 REM      CONTROLLO
1520 REM       A CAPO RIGA
1530 IF LEN(ST$)<LF THEN 1820
1540 L1$=SEG$(ST$,LF-1,1)
1550 L2$=SEG$(ST$,LF,1)
1560 IF PNT<>0 THEN 1580
1570 IF J>UR THEN 1820 ELSE RIGAO$=RIGA$(J
```

```
)
1580 L3$=SEG$(RIGA0$,PNT+1,1):: L4$=SEG$(R
IGA0$,PNT+2,1)
1590 IF L3$=CHR$(32)THEN STS=STS-1 :: GOTO
 1800
1600 IF L2$=L3$ OR L3$=L4$ THEN 1750
1610 IF L1$=L2$ THEN 1790
1620 IF L1$="C" AND L2$="Q" THEN 1790
1630 IF L1$="c" AND L2$="q" THEN 1790
1640 IF L2$=CHR$(32)THEN 1750
1650 PX=POS(",.;:-/+=)%!?'`"&CHR$(34),L2$,
1):: IF PX<>0 THEN 1800
1660 PX=POS(",.;:-/+=)%!?'`"&CHR$(34),L3$,
1):: IF PX<>0 THEN 1750
1670 PX=POS("LRMNlrmn",L1$,1):: IF PX=0 TH
EN 1690
1680 CALL VOCALE(L2$,V):: IF V=0 THEN 1790

1690 CALL VOCALE(L1$,V):: IF V=0 THEN 1750

1700 CALL VOCALE(L2$,V):: IF V=1 THEN 1750

1710 CALL VOCALE(L3$,V):: IF V=1 THEN 1790

1720 IF L3$="R" OR L3$="r" THEN 1790
1730 IF L2$<>"S" AND L2$<>"s" THEN 1750
1740 IF L2$<>L3$ THEN 1790
1750 STS=STS+1 :: IF STS>10 OR STS>LF-1 TH
EN ST$=RIG$ :: GOTO 1820
1760 L4$=L3$ :: L3$=L2$ :: L2$=L1$ :: L1$=
SEG$(ST$,LF-1-STS,1)
1770 ST$=SEG$(ST$,1,LF-STS):: GOTO 1590
1780 GOTO 1820
1790 ST$=SEG$(ST$,1,LEN(ST$)-1)&CHR$(45)::
 STS=STS+1
1800 DIFF=LF-LEN(ST$)
1810 IF DIFF>0 THEN CALL MARGINE(ST$,DIFF)

1820 PRINT #4:RPT$(" ",W);:: IF KK=1 THEN
W=MS
1830 PRINT #4:ST$
1840 IF J>UR THEN 1920
1850 PNT=PNT-STS
1860 IF PNT=28 THEN J=J+1 :: GOTO 1410
1870 IF PNT<1 THEN 1890
1880 RIGA0$=SEG$(RIGA0$,PNT+1,LR0-PNT):: L
R0=LEN(RIGA0$):: GOTO 1910
1890 IF PNT=0 THEN 1410
1900 J=J-1 :: RIGA0$=SEG$(RIGA$(J),LR0+PNT
+1,-PNT):: LR0=LEN(RIGA0$)
1910 PNT=LF :: RIG$="" :: GOTO 1450
1920 CLOSE #4 :: GOTO 250
1930 IF LEN(RIG$)=0 THEN 1970
1940 FOR Q=1 TO LEN(RIG$):: IF ASC(SEG$(RI
G$,Q,1))<>32 THEN PRINT #4:RPT$(" ",W);RIG
$ :: GOTO 1960
1950 NEXT Q :: GOTO 1970
1960 IF KK=1 THEN W=MS
1970 PX2=POS(RIGA$(J),"\",PX1+1):: IF PX2=
0 THEN J=J+1 :: GOTO 1410
1980 COD=VAL(SEG$(RIGA$(J),PX1+1,PX2-PX1-1
)):: IF COD=14 THEN W=W/2 :: KK=1 ELSE KK=
```

il margine sinistro anche in presenza di caratteri ingranditi o condensati.
Siamo convinti che questo Text-Editor, pur non dotato di tutte le funzioni di Writer TI, ne elimina il difetto dovuto alla presenza di finestre laterali ed offre una buona leggibilità accompagnata dalla facilità d'uso. A chi poi adopera attualmente l'Editor/Assembler come elaboratore di testi, dà innegabili vantaggi per la sua capacità di inserimento e cancellazione di caratteri con il riaggiustamento dell'intero capoverso e di controllo del formato stampa. In ogni caso il fiore all'occhiello resta la sua prerogativa di giustificazione del margine destro con la suddivisione delle parole in sillabe secondo la grammatica italiana. Una caratteristica davvero poco comune anche nei word-processor più sofisticati.

**DIFF** Numero di space da inserire in una riga di stampa per giustificare il margine destro.

**COD** Numero di codice per il controllo della stampa.

**FLAG** Variabile flag usata in una subroutine richiamata da un'istruzione On Error; in base al suo valore in programma viene rimandato all'inizio del menu o all'accept del menu stesso.

**FN$** Nome di un file letto dal catalogo del dischetto.

**DN$** Nome del dischetto.

**Principali variabili del programma**

**RIGA$ ()** - Righe di testo di 28 caratteri ciascuna.
**NUMRIGHE** - Numero di righe delle quali è composto il testo in memoria.
**FINA$** - Nome di file in memoria.
**SC$** - Scelta dell'utente tra le operazioni dei menu.
**RECORD$** - Variabile di lavoro per la lettura o la scrittura di un record da o su disco. Ogni record contiene due righe di testo.
**SCHERMO** - Numero dello schermo sul quale è presente una parte di testo; corrisponde al numero della pagina video.
**FINE** - Variabile di lavoro per mostrare la prima pagina di testo.
**RI,CO** - Numero di riga e di colonna sul quale si trova il cursore a forma di quadrato vuoto che si ha nel modo funzioni.
**R$** - Variabile che accetta momentaneamente una riga scritta dall'utente.
**NU** - Numero di riga che si sta editando.
**Z$,W$** - Variabili di lavoro usate prevalentemente per ridefinire le righe dopo un inserimento o la cancellazione di un carattere.
**F1** - Variabile flag che indica la fine del testo o di un capoverso durante la ridefinizione delle righe in seguito alla cancellazione o all'inserimento di un carattere.

```
0
1990 IF COD=15 THEN W=INT(W*33/20)+1
2000 IF COD=18 THEN W=MS
2010 PRINT #4:CHR$(COD);:: PX1=PX2 :: GOTO
 1970
2020 GOSUB 2190 :: IF SC$="Y" OR SC$="y" T
HEN NUMRIGHE=1 :: FINA$="NEW" :: RIGA$(1)=
F$
2030 GOSUB 2390 :: GOTO 300
2040 CALL CLEAR :: ON ERROR 2530 :: OPEN #
3:"DSK1.",INPUT ,RELATIVE,INTERNAL :: CTR=
0 :: T=-7
2050 INPUT #3:DN$,J,J,K :: PRINT "DISKNAME
=";DN$:"available=";K;"used";J-K
2060 PRINT :: FOR LOOP=1 TO 127 :: INPUT #
3:FN$,Y,J,K :: IF LEN(FN$)=0 THEN 2120
2070 IF ABS(Y)=2 THEN PRINT FN$;TAB(19+T);
STR$(J),:: CTR=CTR+1 :: T=-T
2080 IF CTR<36 THEN 2110
2090 PRINT :" PRESS <ENTER> TO CONTINUE"
2100 CALL KEY(0,K,S):: IF S=0 THEN 2100 EL
SE PRINT :: CTR=0
2110 NEXT LOOP
2120 CLOSE #3 :: IF T=7 THEN PRINT
2130 PRINT :"DELETE FILE? (Y/N) N" :: ACCE
PT AT(23,20)SIZE(-1)VALIDATE("YNyn"):R$
2140 IF R$="N" OR R$="n" THEN 250
2150 FLAG=1 :: ON ERROR 2220 :: GOSUB 2390
 :: GOSUB 2200 :: GOSUB 2190
2160 IF SC$="N" AND SC$="n" THEN 250
2170 OPEN #5:FINA$,SEQUENTIAL,INPUT ,DISPL
AY ,VARIABLE 80 :: CLOSE #5:DELETE
2180 FLAG=0 :: FINA$=FNW$ :: GOTO 250
2190 DISPLAY AT(24,6)BEEP:"SEI SICURO? (Y/
N) N" :: ACCEPT AT(24,24)SIZE(-1)VALIDATE(
"YNyn"):SC$ :: RETURN
2200 FNW$=FINA$ :: DISPLAY AT(23,1):"file
name: DSK1." :: ACCEPT AT(23,12)SIZE(-15):
FINA$
2210 RETURN
2220 DISPLAY AT(23,1):"* ERRORE * FILE NOT
 PRESENT" :: DISPLAY AT(24,1):"(PRESS <ENT
ER> TO CONTINUE)"
2230 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 2230 ELSE GOSUB 2390 :: FINA$=FNW$
2240 IF FLAG=1 THEN FLAG=0 :: RETURN 250 E
LSE RETURN 300
2250 GOSUB 2320 :: IF NU<NUMRIGHE THEN RET
URN
2260 IF NUMRIGHE<300 THEN 2290
2270 DISPLAY AT(23,1)BEEP:"* WARNING * OUT
 OF DIMENSION" :: DISPLAY AT(24,1):"premi
 ENTER per continuare"
2280 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 2280 ELSE CALL HCHAR(23,3,32,60):: RI=RI
-1 :: RETURN
2290 RIGA$(NUMRIGHE)=RPT$(" ",28):: NUMRIG
HE=NUMRIGHE+1 :: RIGA$(NUMRIGHE)=F$
2300 DISPLAY AT(RI,1):RIGA$(NUMRIGHE-1)::
IF RI<22 THEN DISPLAY AT(RI+1,1):F$
2310 RETURN
2320 NU=(SCHERMO-1)*20+RI-2 :: RETURN
```

```
2330 IF SCHERMO*20<=NUMRIGHE THEN FINE=22
ELSE FINE=NUMRIGHE+2-(SCHERMO-1)*20
2340 GOSUB 2320 :: Y=0 :: FOR J=RI TO FINE
 :: DISPLAY AT(J,1):RIGA$(NU+Y):: Y=Y+1 ::
 NEXT J
2350 IF SCHERMO=1 THEN GOSUB 2410 :: RETUR
N ELSE NU=(SCHERMO-1)*20 :: DISPLAY AT(1,1
):RIGA$(NU-1):: DISPLAY AT(2,1):RIGA$(NU)

2360 RETURN
2370 NUMRIGHE=NUMRIGHE+1 :: FOR J=NUMRIGHE
 TO NU+1 STEP -1 :: RIGA$(J)=RIGA$(J-1)::
NEXT J
2380 RIGA$(NU)=RPT$(" ",28):: RETURN
2390 CALL HCHAR(23,3,32,60):: RETURN
2400 CALL HCHAR(1,3,32,60):: RETURN
2410 IF FINA$="NEW" THEN W$="edit NEW FILE
" ELSE W$="edit "&FINA$
2420 DISPLAY AT(1,1):W$ :: DISPLAY AT(2,1)
:RPT$("-",28):: RETURN
2430 DISPLAY AT(23,1):"* ERRORE * STAMPANT
E" :: GOTO 2450
2440 DISPLAY AT(23,1):"* ERRORE * CODICI C
ONTROLLO"
2450 DISPLAY AT(24,1):"(PREMI ENTER PER CO
NTINUARE)"
2460 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 2460 ELSE CLOSE #4 :: RETURN 250
2470 DISPLAY AT(24,1):"* ERRORE * PRESS <E
NTER>."
2480 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 2480 ELSE RETURN 840
2490 RETURN 300
2500 PX=POS(R$,",",1):: IF PX=0 THEN R1=VA
L(R$):: R2,R3=0 :: RETURN
2510 R1=VAL(SEG$(R$,1,PX-1)):: PY=POS(R$,"
,",PX+1):: IF PY=0 THEN R2=VAL(SEG$(R$,PX+
1,LEN(R$)-PX)):: R3=0 :: RETURN
2520 R2=VAL(SEG$(R$,PX+1,PY-PX-1)):: R3=VA
L(SEG$(R$,PY+1,LEN(R$)-PY)):: RETURN
2530 PRINT " * ERROR * DISK NOT PRESENT":"
PRESS <ENTER> TO CONTINUE."
2540 CALL KEY(0,K,S):: IF S=0 OR K<>13 THE
N 2540
2550 ON ERROR 2560 :: CLOSE #3
2560 RETURN 250
2570 !@P+
2580 END
2590 SUB VOCALE(LX$,V)
2600 V=POS("AEIOUaeiou",LX$,1):: IF V<>0 T
HEN V=1
2610 SUBEND
2620 SUB MARGINE(RX$,W)
2630 LS=LEN(RX$):: CTR=0
2640 FOR J=1 TO LS-1 :: L=ASC(SEG$(RX$,LS-
J,1))
2650 IF L=32 THEN RX$=SEG$(RX$,1,LS-J)&" "
&SEG$(RX$,LS-J+1,J+CTR):: CTR=CTR+1
2660 IF CTR=W THEN SUBEXIT
2670 NEXT J :: IF CTR<W THEN W=W-CTR :: GO
TO 2630
2680 SUBEND
```

**R1,R2,R3** - Memorizzano rispettivamente la prima e l'ultima riga di un blocco di testo e il numero di riga oltre il quale va collocato. Le variabili sono usate con le funzioni: Move, Copy, Insert, Delete.

**BLOCCO** - Lunghezza, espressa in numero di righe, del blocco di testo da inserire o da copiare!

**CTR** - Variabile con funzione di contatore.

**PX** - Posizioni in cui si trovano le virgole che separano i numeri di riga specificati dopo una funzione Move, Copy, ecc.

**FNW$** - Variabile usata per memorizzare momentaneamente il nome di un file.

**DEV$** - Nome dell'interfaccia per la stampante.

**MS** - Numero di colonne lasciate oltre il margine sinistro nel testo stampato.

**LF** - Larghezza del formato di stampa.

**UR** - Numero di righe da stampare.

**PNT** - Variabile di lavoro per formattare le righe per la stampa.

**RIG$** - Riga di LF caratteri prima del controllo per andare a capo riga nel modo grammaticalmente corretto.

**PX1, PX2** - Posizioni delle barre oblique inverse nelle stringhe contenenti i codici di controllo per la stampa.

**RIGAO$** - Variabile di lavoro usata per formattare le righe per la stampa.

**LRO** - Lunghezza di una riga nel formato video.

**ST$** - Riga nel formato

stampa.
**TS** - Numero di caratteri di cui deve essere ridotta la riga nel formato di stampa per poter andare a capo correttamente. la lunghezza LF viene poi ripristinata distribuendo un uguale numero di space all'interno della riga.
**L1$,L2$** - Penultimo ed ultimo carattere di una riga nel formato stampa.
**L3$,L4$** - Primo e secondo carattere della riga successiva a quella che dev'essere stampata. La loro conoscenza è indispensabile per poter andare a capo in modo corretto.
**V** - Variabile flag che indica la presenza di una vocale.

*Tratto da Personal Software N° 27/1985.*

# ISTOGRAMMI A BARRE

Con questo programmino in Extended Basic sarete in grado di visualizzare sullo schermo l'andamento di qualsiasi funzione (dal profitto della vostra azienda alla media scolastica), con una completa presentazione grafica. Il programma vi chiederà il titolo eventuale che intendete dare al grafico, il numero di valori che volete inserire, e se desiderate avere delle didascalie sull'asse orizzontale. Dopo aver inserito tutti i valori e le eventuali didascalie, il TI 99 vi presenterà il grafico calcolandone automaticamente la scala di rappresentazione e ponendo in cima ad ogni barra il relativo valore numerico.
Premere un tasto per inserire nuovi valori.
*Tratto dall'archivio Jackson*

```
100 CALL CLEAR :: OPTION BASE 1 :: CALL SC
REEN(2):: FOR I=5 TO 14 :: CALL COLOR(I,11
,2):: NEXT I :: FOR I=2 TO 4 :: CALL COLOR
(I,8,1):: NEXT I
110 DIM J(27),COST(27),S$(27,6),D$(27):: C
ALL CHAR(143,"007E7E7E7E7E7E7E")
120 CALL CHAR(97,"00000000000000007E00000000
00007E7E00000000007E7E7E000000007E7E7E7E")

130 CALL CHAR(101,"0000007E7E7E7E00007E7
E7E7E7E7E007E7E7E7E7E7E7E7E7E7E7E7E7E7E7E"
```

```
)
140 FOR I=1 TO 27 :: J(I)=0 :: COST(I)=0 :
: D$(I)="" :: FOR K=1 TO 6 :: S$(I,K)="" :
: NEXT K :: NEXT I
150 PRINT "PRG.ISTOGRAMMI": : : : : : : :
: : : : : : : : : : : :: PRINT "TITOLO? (MAX
 20 CARATT)": : :: ACCEPT AT(24,1)BEEP SIZ
E(20):C$
160 INPUT "QUANTI VALORI?(MAX 27) ":N :: I
NPUT "DIDASCALIE?(MAX 3 CARATTERI)":A$ ::
IF A$="SI" THEN KNST=1 ELSE KNST=0
170 FOR I=1 TO N :: PRINT :"DATO N.";I;::
INPUT J(I):: IF KNST=1 THEN INPUT "DIDASCA
LIA? ":D$(I)
180 GOSUB 400 :: IF J(I)>16 THEN COST(I)=1
 ELSE COST(I)=0
190 NEXT I :: CALL CLEAR :: GOSUB 380 :: D
ISPLAY AT(1,5)SIZE(20):C$ :: FOR I=1 TO N
:: IF COST(I)=1 THEN GOSUB 340 :: GOTO 210

200 NEXT I
210 FOR F=1 TO N :: A=F+2 :: IN=INT(J(F)):
: B=22-IN :: CALL VCHAR(B,A,143,IN):: GOSU
B 240 :: GOSUB 410 :: IF KNST=1 THEN GOSUB
 440
220 NEXT F
230 CALL KEY(0,K,S):: IF S=0 THEN 230 ELSE
 140
240 DEC=(J(F)-IN)*10 :: IF DEC<=.99 THEN R
ETURN
250 ON INT(DEC)GOTO 260,270,280,290,300,31
0,320,330,330
260 CALL HCHAR(B-1,A,97):: RETURN
270 CALL HCHAR(B-1,A,98):: RETURN
280 CALL HCHAR(B-1,A,99):: RETURN
290 CALL HCHAR(B-1,A,100):: RETURN
300 CALL HCHAR(B-1,A,101):: RETURN
310 CALL HCHAR(B-1,A,102):: RETURN
320 CALL HCHAR(B-1,A,103):: RETURN
330 CALL HCHAR(B-1,A,104):: RETURN
340 X=.99
350 FOR I=1 TO N :: J(I)=J(I)*X :: NEXT I
:: FOR I=1 TO N :: IF J(I)>16 THEN 370
360 NEXT I :: RETURN
370 X=X-.01 :: GOTO 350
380 CALL CHAR(120,"FF"):: CALL CHAR(121,"0
101010101010101"):: CALL HCHAR(22,3,120,28
):: CALL VCHAR(3,2,121,18):: CALL HCHAR(1,
3,120,28)
390 CALL VCHAR(3,30,121,18):: RETURN
400 A$=STR$(J(I)):: FOR K=1 TO LEN(A$):: S
$(I,K)=SEG$(A$,K,1):: NEXT K :: RETURN
410 H=B-5 :: IF H<1 THEN H=1
420 FOR K=1 TO 6 :: IF S$(F,K)="" THEN RET
URN ELSE DISPLAY AT(H,A-2)SIZE(-1):S$(F,K)

430 H=H+1 :: NEXT K :: RETURN
440 R=22
450 FOR I=1 TO 3 :: DISPLAY AT(R,A-2)SIZE(
-1):SEG$(D$(F),I,1):: R=R+1 :: NEXT I :: R
ETURN
```

# i Tascabili

Programmi di statistica — Boris Allan
La programmazione — John Shelley
WordStar — Maddie Labinger
Assembler 6502 — Bob Bright
COBOL — Ray Welland
VIC 20 — Ron Geere
UNIX — Lawrence Blackburn, Marcus Taylor
BASIC — Roger Hunt
PASCAL — David Watt
FORTRAN 77 — Philip Ridler
Commodore 64 — Boris Allan
Sharp MZ-80A — Roger Morton
PC-DOS — Val King, Dick Waller
MS DOS — Val King e Dick Waller
PC IBM — Peter Gosling

# Jackson, naturalmente.

## ELENCO DEI TITOLI DISPONIBILI

I tascabili Jackson sono uno strumento prezioso per chi lavora con il computer.

- SINCLAIR SPECTRUM cod. 017H
- VIC 20 cod. 005H
- COMMODORE 64 cod. 002H
- PC IBM cod. 018H
- APPLE IIc cod. 003H
- SHARP MZ80A cod. 014H
- LA PROGRAMMAZIONE cod. 004H
- WORD STAR cod. 008H
- UNIX cod. 009H
- LOGO cod. 020H
- MS-DOS cod. 019H
- PROGRAMMI DI STATISTICA cod. 015H
- CP/M cod. 011H
- PC-DOS cod. 012H
- BASIC cod. 007H
- ASSEMBLER Z80 cod. 016H
- ASSEMBLER 6502 cod. 013H
- COBOL cod. 001H
- FORTRAN 77 cod. 010H
- PASCAL cod. 006H

**OGNI TASCABILE COSTA L. 8.500**

ritagliare (o fotocopiare) e spedire in busta chiusa a:
**GRUPPO EDITORIALE JACKSON** - Divisione Libri - Via Rosellini, 12 - 20124 Milano

### CEDOLA DI COMMISSIONE LIBRARIA

**VOGLIATE SPEDIRMI**

| n° copie | codice | Titolo | Prezzo unitario | Prezzo totale |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | Totale | |

☐ Pagherò contrassegno al postino il prezzo indicato più **L. 3.000 per contributo fisso spese di spedizione.**

Condizioni di pagamento con esenzione del contributo spese di spedizione:

☐ Allego assegno della Banca ☐ Allego fotocopia del versamento sul c/c n. 11666203 a voi intestato

N° ☐ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome e Cognome

Via

Cap Città Prov.

Data Firma

Spazio riservato alle Aziende Si richiede l'emissione di fattura

Partita I.V.A.

ORDINE MINIMO L. 50.000